

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

RECONOCIMIENTO DE CALIDEZ OFICIAL, ACUERDO SEP. NO. 15018
PUBLICADO EN EL DIARIO OFICIAL DE LA FEDERACIÓN
EL 29 DE NOVIEMBRE DE 1976.

DEPARTAMENTO DE ELECTRÓNICA, SISTEMAS E INFORMÁTICA
MAESTRÍA EN INFORMÁTICA APLICADA



ITESO

**APP: SISTEMA DE REPUTACIÓN EN MATERIA DE GASOLINERAS
EN MÉXICO.**

PROYECTO PROFESIONALIZANTE DE DESARROLLO TECNOLÓGICO PARA
OBTENER EL
GRADO DE MAESTRO EN INFORMÁTICA APLICADA

PRESENTA:

Christian Humberto Rojo de la Cruz

Asesora del trabajo de obtención de grado
Mtra. Sara Josefa Ortiz Cantú

Contenido

RESUMEN	3
1. INTRODUCCIÓN	4
2. PROBLEMA.	5
3. HIPOTESIS.	8
4. OBJETIVOS.	9
4.1 OBJETIVO GENERAL.	9
4.2 OBJETIVOS PARTICULARES.	9
5. JUSTIFICACIÓN.....	10
6. MARCO TEÓRICO.....	13
6.1 SISTEMAS DE REPUTACIÓN.	13
6.1.1 Arquitectura de Red de Sistemas de Reputación.....	15
6.1.1.1 Sistemas de Reputación Centralizados.	16
6.1.1.2 Sistemas de Reputación Distribuidos.	17
6.1.2 Motores de cálculo de Reputación.	18
6.1.3 Retos en los Sistemas de Reputación.....	20
6.2 INGENIERÍA DE SOFTWARE.	22
6.2.1 Dominios de aplicación del Software.	22
6.2.2 Ingeniería de software.	24
2.2.1 Proceso de la Ingeniería de software.	25
2.2.1.1 Modelo de proceso del software.	25
2.2.3 Métodos de la ingeniería del software.	29
6.3 APLICACIONES MÓVILES.	30
6.3.1 Métodos aplicados al desarrollo de aplicaciones móviles	33
6.3.1.1 Modelo de cascada.....	34
6.3.1.2 Desarrollo rápido de aplicaciones	34
6.3.1.3 Desarrollo ágil.....	34
6.3.1.4 Mobile-D.....	36
6.3.2 Categorías de aplicaciones móviles:.....	37
6.3.3 Monetización de apps	39
6.3.3.1 Modelos de monetización.....	41
7. DESARROLLO DE LA APLICACIÓN.	43

7.1 ANÁLISIS.	44
7.1.1 Obtención y clasificación de requerimientos.....	44
7.1.2 Propuesta de solución.....	47
7.2 DISEÑO.	48
7.2.1 Definir escenarios.....	48
7.2.2 Arquitectura del software	50
7.2.3 Diseño/plot de la secuencia de la App.	51
7.3 DESARROLLO.	53
7.3.1 Creación de la base de datos y Servicios Web.	53
7.3.2 Creación de pantallas y codificación de la App.	58
7.4 PRUEBAS.....	61
7.4.1 Pruebas Unitarias.	61
7.4.2 Pruebas Integrales.....	63
7.5 LIBERACIÓN DEL APK/RESULTADOS.....	65
8. CONCLUSIONES.	67
BIBLIOGRAFÍA.....	71
ANEXOS.	74

RESUMEN

En el siguiente documento se muestra mi proyecto profesionalizante de desarrollo tecnológico para la obtención de grado, que consistió en la elaboración de una aplicación móvil para poder crear un sistema de reputación sobre gasolineras en México.

Dentro de este trabajo se exponen los motivos que me llevaron a tomar este proyecto, así como la base teórica y el detalle del desarrollo de la aplicación.

La codificación de esta app se hizo entre Septiembre 2015 y Junio 2016 por lo que desde el término de su desarrollo han cambiado algunas cosas con respecto al contexto con el que se hizo la propuesta de este trabajo:

- El precio de las gasolinas que se daba mensualmente por parte del gobierno, ya no es fijo en todo México, puede cambiar diariamente y ser distinto dependiendo el municipio.
- Apertura en materia de comercializadores de combustibles, PEMEX ya no es la única empresa que puede vender gasolina y diésel en México, otras marcas compañías pueden hacerlo.
- Salida al mercado de aplicaciones que tienen prácticamente las mismas funciones que las que este trabajo propone.

1. INTRODUCCIÓN

Según datos del INEGI en 2014 existían 38,023,535 vehículos de motor registrados en circulación en México (entre automóviles, camiones, motos) (INEGI, 2014) y esta cifra aumenta en 2 millones aproximadamente, año con año. La gran mayoría de estos vehículos utilizan como combustible gasolina (ya sea magna o Premium) y diésel, éstas se surten a lo largo del país por una sola empresa de combustible PEMEX, que permite terceros (concesionarios) ofrezcan estos combustibles a los usuarios finales (automovilistas), en lo que conocemos como gasolineras.

De acuerdo con las cifras en el Anuario Estadístico 2014 de PEMEX, en México se contaba con 10,820 estaciones de servicio a lo largo de todo el territorio nacional, con una venta de 1,112.6 miles de barriles por día (entre gasolina y diésel) durante 2014, lo que representó 622,556.9 millones de pesos durante ese año (PEMEX, 2014).

Estos datos muestran lo grande que es este mercado y el volumen de dinero y producto que se maneja día con día, pero ¿qué sucede si los proveedores de estos combustibles tuvieran la capacidad y la intención de modificar sus máquinas despachadoras para entregar menos cantidad de combustible al comprador, y cobrarle como si le hayan despachado correctamente?, sería una gran cantidad de dinero en México la que se estuviera fugando por esta práctica.

Al menos en gran parte de la población existe esa percepción, que en algunas estaciones de servicio no se despacha realmente la cantidad de combustible que se está cobrando.

La propuesta de este proyecto es realizar un sistema de reputación para que la usuarios de gasolineras tengan la oportunidad de evaluar el servicio que están recibiendo, a este sistema podrían acceder los usuarios mediante una aplicación móvil instalada en su celular.

Este sistema permitiría al usuario compartir su experiencia (podría subir comentarios, imágenes, etc.) a la hora de cargar gasolina en determinada estación (geolocalización) y poder valorarla (*rating*), con lo que se estaría generando una base de información que ayudaría a otros usuarios a tomar una mejor decisión a la hora de elegir una estación donde cargar gasolina.

También por otra parte, podría servir para exponer a las estaciones de servicio que estén otorgando un mal servicio o los usuarios consideren que no están entregando la cantidad de combustible que están cobrando, y de esta manera forzarlos a que se regularicen.

La manera más adecuada y funcional para llevar a cabo este propósito es a través de una aplicación móvil, debido a que el Smartphone se ha convertido en un elemento de nuestra vida cotidiana y es de muy fácil acceso para compartir cosas desde alguna ubicación en específico (servicios de geolocalización) (Shankar et al. 2012).

La comunidad al estar participando en tiempo real y dar su retro, estaría trabajando como un sensor de una realidad (en este caso la percepción sobre el llenado de combustible en las distintas estaciones de servicio) (Mousa et al. 2015), y de esta manera formando un conocimiento público, para ayudar a la misma comunidad (Parera et al. 2014).

2. PROBLEMA.

En México el precio de la gasolina está controlado por el gobierno, así que el mismo tipo de combustible debe costar lo mismo en cualquier gasolinera donde se pretenda surtir, lo que resulta complicado para los usuarios es que no se tiene certeza de que en las gasolineras te pongan la cantidad de combustible que dicen que te están poniendo (y sobre la cual estás pagando), lo cual es un tema importante para muchas personas, ya que el precio de la gasolina ha subido bastante en los últimos años y es un producto que se sigue necesitando sin importar cuál sea su precio (demanda inelástica).

El gasto en gasolina depende de los desplazamientos de cada persona, tipo de automóvil y hábitos de manejo, pero a todos les afecta, ya que es un gasto fuerte que se hace recurrentemente y no se desea estar pagando de más cada vez que se hace una carga de gasolina o recibir menos de lo pagado.

En el contexto mexicano, PROFECO es la entidad encargada de revisar que las bombas despachadoras de combustible en las gasolineras estén calibradas y cuenten con los hologramas de verificación de la Unidad de Verificación Aprobada (UVA), además de que no hayan sufrido alguna modificación en su software o hardware, esto para asegurar que las bombas despachadoras de las gasolineras surtan “litros de litro”, y cuando se detectan irregularidades en el funcionamiento de estas bombas, se puede proceder a su inmovilización y aplicación de multas a los concesionarios.

Esta revisión la hace de manera sorpresiva y aleatoria, iniciando el proceso de verificación cuando el jefe de una brigada de la Profeco recibe la orden de inspección el mismo día en que realizará la visita (y no antes, con el propósito de evitar fugas de información que prevengan a la estación de servicio que será inspeccionada), una vez que la brigada llega a la gasolinera a revisar, se identifica y delante de testigos procede a hacer su revisión.

Profeco revisa diferentes aspectos de la estación de servicio, pero destina parte importante de la verificación a constatar que las bombas despachadoras de combustible funcionen correctamente.

La brigada asegura, en primer lugar, que cada bomba cuente con el holograma de verificación vigente. Después realiza la verificación volumétrica, revisa que la manguera despache litros de a litro. Para ello utiliza unos recipientes llamados “jarras” (recipientes de 20 litros que tienen una escala perfectamente graduada para detectar anomalías cuando se despacha la gasolina o el diésel).

Se llevan a cabo nueve mediciones por cada manguera verificada, tres en la velocidad más baja en que puede despachar, tres a velocidad media y tres más a la velocidad más alta, para detectar “errores de repetibilidad”. Estos se presentan

cuando una bomba despacha distintas medidas de combustible para la misma cantidad solicitada.

Según la normatividad vigente NOM-005-SCFI-2011 se puede tener un faltante de hasta 100 mililitros por cada 20 litros de combustible, esto debido a que puede haber desgaste en las bombas por uso o por la temperatura del medio ambiente. Las mangueras que rebasan este margen de error son inmovilizadas. El error de repetibilidad se presenta cuando el margen de dispersión entre los resultados obtenidos en tres despachos consecutivos de 20 litros supera los 60 mililitros (Secretaría de Economía, 2012).

Esta causal de inmovilización (cualidades metrológicas) es sólo una de las veinticuatro causales de inmovilización de una manguera (agrupadas en 9 grupos: acreditación documental, calidad del combustible, cualidades metrológicas, electrónicas, fallas hidráulicas, falta o deficiencia en servicio de calibración, fuera de servicio, precio y seguridad), pero es la más común en las verificaciones que realiza Profeco (PROFECO, 2016). En 2014, la mitad de las mangueras paralizadas presentaron alguna de estas fallas.

En la región conformada por Jalisco, Colima y Michoacán, las irregularidades alcanzaron a más del 70% de las estaciones de servicio verificadas. Las fallas también se hicieron presentes en siete de cada 10 gasolineras inspeccionadas en Oaxaca, Zacatecas y Nuevo León.

Es muy complicado para Profeco vigilar todas las gasolineras. En 2014 se realizaron 1,800 operativos únicamente. Aunado a esto en algunas zonas, en especial aquellas con fuerte presencia del crimen organizado (Michoacán, Tamaulipas y Guerrero) la dependencia se limitó a revisar menos del 10% de las gasolineras.

En Michoacán y Tamaulipas apenas cubrió el 3% de las estaciones de servicio, mientras que en Guerrero revisó el 9%. Inclusive en la capital del país, solamente se verificó 45% de los concesionarios.

Por estos datos anteriores se puede decir que en México existe un problema real de variación, entre lo que se cobra en las estaciones de servicio y lo que realmente se despacha, y Profeco no puede cubrir de una manera adecuada todas las gasolineras, ya sea por limitaciones de personal o por lugares difíciles de acceder (en buena medida por la presencia del crimen organizado).

3. HIPOTESIS.

Como se menciona en el marco teórico uno de los factores que más influye en la decisión de los consumidores a la hora de elegir algún producto o servicio es el estatus o reputación de éstos.

Estos sistemas de reputación en la actualidad además de los canales tradicionales “boca a boca”, pueden ser más amplios, incluso llegando a dimensiones globales gracias al internet, donde existen comunidades online de reseñas (*reviews*) acerca de productos y servicios.

La meta final de estos sistemas es ayudar a una entidad (persona o empresa), dentro de una comunidad diversa, a elegir otra entidad confiable para interactuar con ella (Koutrouli & Tsalgatiidou 2015). Para lograr esta meta lo hacen normalmente a través de valoraciones y comentarios de usuarios que previamente ya utilizaron el producto o servicio que se está valorando.

Estos sistemas pueden ser administrados por la misma compañía para ayudar y dar confianza a sus compradores a la hora de realizar sus transacciones (ebay, amazon), o pueden ser administrados por una empresa sin ningún tipo de relación con las empresas proveedoras de productos y servicios que se están valorando (Yelp), cuya única finalidad es que los consumidores de estos productos y servicios tomen la mejor decisión.

Un buen número de investigaciones han teorizado y empíricamente validado el efecto significativo que tienen los *online reviews* en las decisiones de compra de los consumidores y en las utilidades de los negocios (Duan et al., 2008; Wangenheim & Bayón, 2007) .

Considero que un sistema de reputación podría ayudar a mitigar el problema que se ha expuesto anteriormente sobre la diferencia que existe entre el cobro y lo que realmente despachan las estaciones de servicio de combustible (gasolineras) en México, debido a que no alcanza el trabajo que hace Profeco, en verificar las diferentes estaciones, ya que son demasiadas y algunas no son fácilmente accesibles.

4. OBJETIVOS.

4.1 OBJETIVO GENERAL.

Creación de un sistema de reputación sobre gasolineras en México, utilizando una aplicación móvil, para que los usuarios puedan compartir sus experiencias y valoraciones al cargar gasolina en una estación en específico.

4.2 OBJETIVOS PARTICULARES.

- Permitir a los usuarios tomar una mejor decisión a la hora de elegir una estación donde cargar gasolina, gracias a la base de información que los mismos usuarios estarían generando.
- Exponer a las estaciones de servicio que estén otorgando un mal servicio o los usuarios consideren que no están entregando la cantidad de combustible que están cobrando, y de esta manera forzarlos a que se regularicen.
- Conocer el sentir de los usuarios después de utilizar la aplicación, mediante la generación de un reporte de resultados de las pruebas para ver si se cumple con los objetivos propuestos.

5. JUSTIFICACIÓN.

Como ya se mencionó, el tamaño mercado de la gente que consume gasolina y diésel en México es muy grande (una venta de 1,112.6 miles de barriles por día, entre gasolina y diésel, durante 2014), y día con día aumenta más.

Es un producto básico y necesario para gran parte de la población, y un cambio en el precio o en el rendimiento de este producto tiene gran impacto.

Ante la imposibilidad de respuesta rápida y completa por parte de los verificadores oficiales de las estaciones de servicio para comprobar que las estaciones no están cometiendo algún tipo de irregularidad a la hora de despachar combustible, es necesario encontrar otras vías para que los usuarios puedan tener confianza a la hora de cargar combustible, sobre lo que están pagando es realmente lo que está entrando a sus tanques de combustible.

Por eso se propone crear un sistema de reputación de gasolineras, alimentando por los mismos usuarios, y esto les permita elegir la mejor opción (o la opción más confiable) a la hora de cargar gasolina, así como también exponer a las estaciones que tengan malas prácticas.

En la actualidad ya intentos tecnológicos que se han hecho para ayudar a los usuarios a identificar los lugares (gasolineras) donde se ponen la cantidad de combustible de manera correcta:

- El Universal tiene una página dentro de su dominio llamada litros completos <http://litroscompletos.eluniversal.com.mx/> , que tiene como base la información del *Programa Nacional de Verificación de Combustibles “¿Quién es quién en los combustibles?”* de Profeco (PROFECO, 2016), esta información está actualizada al 27 de noviembre 2015 y muestra en un mapa (de Google Maps) las gasolineras que están en esta base de datos y si en esta revisión presentaron alguna anomalía, semaforizandolas dependiendo del tipo de anomalía (verde: sin anomalías; roja: con anomalías; amarillo: sin verificar y naranja: negó verificación)

Fig 1.

Para ingresar a esta información es vía Web, pero el Universal en 2016 lanzó la APP para consultar esta información desde el móvil.

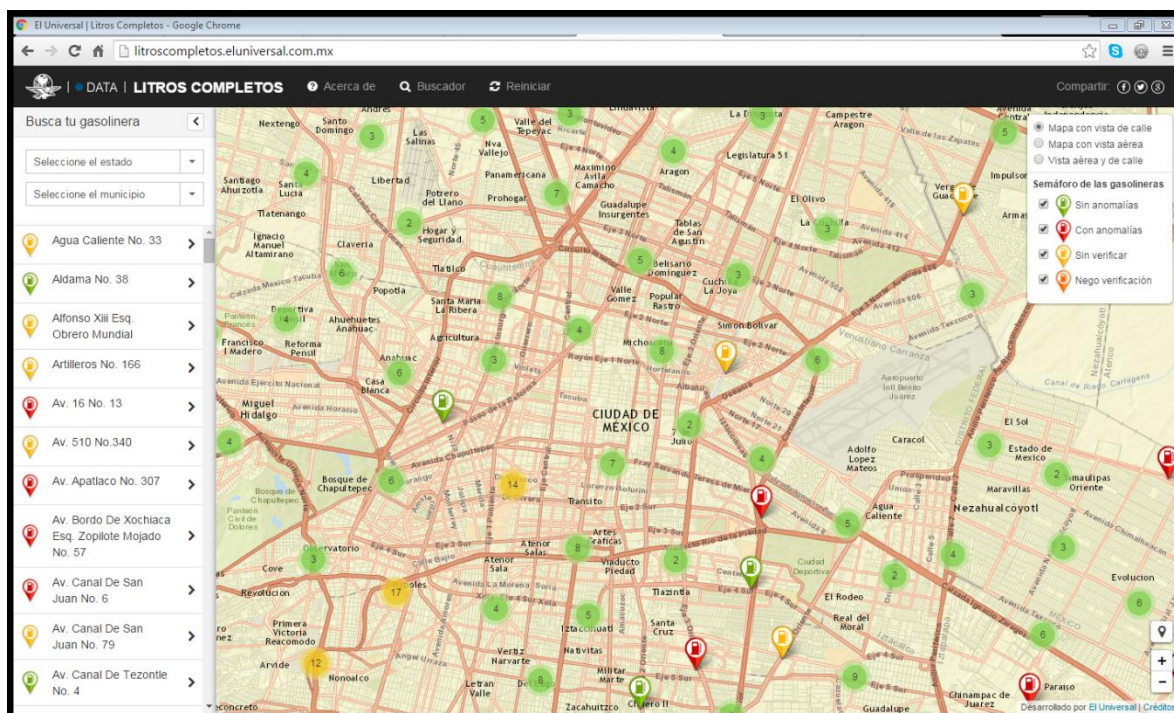


Fig 1. Captura de pantalla tomada desde <http://litroscompletos.eluniversal.com.mx/>

Existe una empresa Mexicana llamada Zenzzer (<http://zenzzer.com/>), que es una combinación de una app con un dispositivo que se pone en los automóviles que mide la cantidad de combustible que realmente te despachan y lo manda a una aplicación móvil con la cantidad de combustible que realmente ingresó al tanque (esta aplicación y dispositivo aún no sale al mercado, están en busca de inversionistas y distribuidores).

Tabla 1. Aplicaciones orientadas a informar sobre el despacho de gasolina en las estaciones.

Aplicación	Descripción	Fortalezas/Debilidades	Dif. vs nueva app
Litros Completos Dev: El Universal CPN México	Muestra la información de su página WEB, que son los datos de las verificaciones que hace Profeco, y muestra un semáforo de las gasolineras con irregularidades.	Fortalezas • Buen diseño. • Ubicación GPS. Debilidades • Contiene únicamente los datos de Profeco (gran porcentaje de estaciones no verificadas).	<ul style="list-style-type: none"> • La app propuesta mostrara info de todas las estaciones, no sólo las verificadas por Profeco.

trueGas - Gasolineras Confiables. Dev: Ruben Maya México	Muestra dentro de un Mapa de Google Maps la ubicación de las diferentes estaciones de servicio, y al darle click te muestra si ha tenido alguna anomalía según los datos de Profeco.	Fortalezas • Ubicación GPS. Debilidades • Datos solamente de Profeco. • Pocas gasolineras, sólo 1,500. • Diseño deficiente. • No es visual el semáforo de las gasolineras.	• Se mostrarán más gasolineras y los semáforos serán más claros para el usuario.
Zenzzer Dev: Zenzzer México	Es un dispositivo que se instala en el coche y mide en tiempo real la cantidad de combustible que se está cargando, y esta información puede ser visualizada desde algún dispositivo móvil.	Fortalezas • Mediciones exactas de la carga de combustible. • Propuesta que se hizo viral en diferentes medios. Debilidades • Necesidad de instalar un dispositivo para realizar la medición. • Aún no sale al mercado.	• No necesitará un dispositivo extra para que funcione.

Fuente: Elaboración propia 2016

- Existen aplicaciones de ciertos grupos de varias estaciones para su ubicación o facilidad de trámites dentro de estos Tabla 2.

Tabla 2. Aplicaciones de Grupos Gasolineros.

Aplicación	Descripción	Fortalezas/Debilidades	Dif. vs nueva app
Petro Seven Dev: Xoles.com México	Muestra la ubicación de estaciones de servicio pertenecientes a este grupo, y el tipo de combustible que manejan y los negocios que se encuentran en esta estación así como otras utilidades para los usuarios.	Fortalezas • Varias utilidades extra para el usuario. Debilidades • No funciona actualmente la APP. • Sólo contiene las gasolineras del grupo.	• No contiene ranking y sólo muestra las gasolineras de ese grupo.
Gasmart Móvil Dev: Gasmart	Muestra la ubicación de estaciones de servicio pertenecientes a este grupo, permite hacer consultas y reportes por flotilla para clientes empresariales.	Fortalezas • Funciones para clientes empresariales. Debilidades • Sólo contiene las gasolineras del grupo.	• No contiene ranking y sólo muestra las gasolineras de ese grupo.
GasoreD Dev: Carlos Tames	Muestra la ubicación de estaciones de servicio	Fortalezas • Información detallada	• No contiene ranking y sólo

Mosino	pertenecientes a este grupo y los tipos de servicio con los que cuenta la estación.	sobre sus estaciones. Debilidades • Sólo contiene las gasolineras del grupo.	muestra las gasolineras de ese grupo.
--------	---	--	---------------------------------------

Fuente: Elaboración propia 2016

-Hay otras apps en Google Play e iTunes Store, enfocadas al registro de consumo de tu vehículo, donde pones cuanto combustible estás cargando y queda registrado para sacar estadísticas de dinero gastado mensualmente, rendimiento por kilómetro, etc.

- También existen Aplicaciones enfocadas a informar al usuario en tiempo real los precios de la gasolina en las diferentes estaciones de servicio cerca de su ubicación (esto aplica en países donde existen diferentes compañías para proveer combustible y su precio puede ser variable).

6. MARCO TEÓRICO

Dentro de este capítulo se desarrollaran los temas que servirán de sustento al proyecto del desarrollo de la aplicación móvil, los tres grandes temas que se abordaran en esta sección son: sistemas de reputación, ingeniería de software y desarrollo de aplicaciones móviles.

6.1 SISTEMAS DE REPUTACIÓN.

Una de los factores más influyen en la decisión de los consumidores a la hora de elegir algún producto o servicio desde siempre ha sido el estatus o reputación del producto o servicio en sí, o de la compañía que los provee, tradicionalmente esta reputación es construida por la recomendaciones de los usuarios de estos productos y/o servicios en lo que se conoce como el “boca a boca” (*“word of mouth” WOM*).

En la actualidad estos canales tradicionales de “boca a boca”, pueden ser más amplios, incluso llegando a dimensiones globales gracias al internet, “boca a boca

electrónico” (*eWOM*), donde existen comunidades online de reseñas (*reviews*) acerca de productos y servicios.

La recopilación sistemática y presentación de una manera estructurada y medible de esta reputación o confianza, para apoyar la decisión de los usuarios de estos productos y/o servicios, es lo que se conoce como sistemas de reputación (Jøsang, et al., 2007).

Aunque existen diferentes métodos para implementar estos sistemas, su meta final es ayudar a una entidad (persona o empresa), dentro de una comunidad diversa, a elegir otra entidad (persona o empresa) confiable para interactuar con ella (Koutrouli & Tsalgatidou, 2015).

De acuerdo con Resnick et al. (2000) los sistemas de reputación tienen las siguientes tres propiedades:

- Se espera que las entidades interactuantes existan en el futuro, por cada interacción actual hay expectativa de una posible interacción en el futuro.
- Ratings (valoraciones) acerca de las interacciones actuales son capturadas y distribuidas.
- Ratings acerca de las interacciones pasadas ayudan guiar las decisiones de las interacciones actuales.

Para proveedores de productos o servicios estos sistemas de reputación representan una herramienta de marketing, mientras que para los consumidores de estos productos y servicios, representa un mecanismo social de moderación y control, así como un método de mejorar la calidad de los mercados y las comunidades (Jøsang & Golbeck, 2009).

Este sistema puede ser administrado por la misma compañía para ayudar y dar confianza a sus compradores a la hora de realizar sus transacciones (ebay, amazon), o puede ser administrado por una empresa sin ningún tipo de relación con las empresas proveedoras de productos y servicios que se están valorando (Yelp), cuya única finalidad es que los consumidores de estos productos y servicios tomen la mejor decisión.

Yelp es un ejemplo exitoso de estos sistemas de reputación, esta compañía estadounidense fue fundada en 2004 por dos ex empleados de Paypal, donde convertían ese “boca a boca” informal, en un “boca a boca electrónico” (*eWOM*), poniendo a disposición de sus usuarios (también llamados *yelpers*) una plataforma electrónica donde pudieran compartir sus experiencias (*crowdsourcing*), con otros usuarios, al utilizar determinados productos y servicios, especialmente de negocios locales (Ma et al., 2014 p. 280). En la actualidad esta compañía cuenta con presencia en 27 países y cotiza en la bolsa de Estados Unidos.

La Fig.2 muestra como los sistemas de reputación y confianza (TRS por sus siglas en inglés) están funcionalmente integrados con la interacción de sus *partners* en un mercado o una comunidad. Una entidad de servicio es asumida como un proveedor confiable por otra entidad o entidades (*relying parties*) que depositan su confianza en ella. Las *relying parties* se asume que confían en los *scores* de su servicio elegido y son confiables para dar retroalimentación o ratings de las entidades de servicio, para alimentar el TRS.

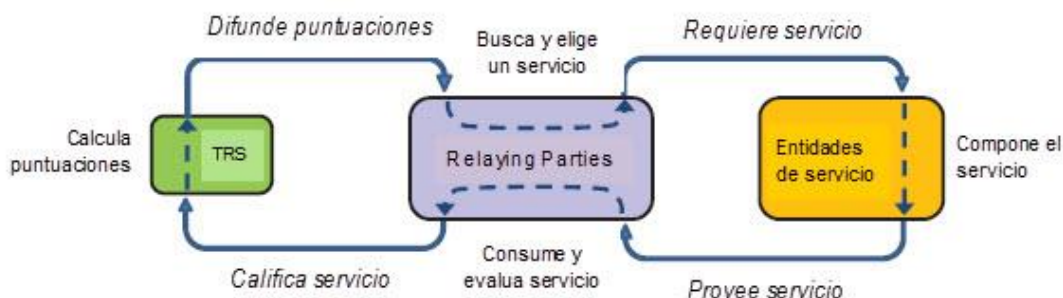


Fig. 2 Integración funcional de un sistema de reputación (Jøsang & Golbeck, 2009 p 2)

Esta figura es una vista funcional de los sistemas de reputación, no muestra una arquitectura debido a que es posible que la función del sistema de reputación esté repartida entre las *relying parties* en un sistema de información de tipo P2P.

6.1.1 Arquitectura de Red de Sistemas de Reputación

La arquitectura de red determina como los ratings y las calificaciones (*scores*) van a ser comunicadas entre los participantes de un sistema de reputación. Las

dos principales clases que existen son: arquitectura centralizada y arquitectura distribuida.

6.1.1.1 Sistemas de Reputación Centralizados.

En un sistema de reputación centralizado, la información acerca del desempeño de un participante es recolectada como ratings de otros miembros de la comunidad que han tenido experiencia de interacción directa con este participante. La autoridad central, que recolecta todos estos ratings determina el puntaje de reputación y confianza de los participantes y los hace públicos para que puedan ser consultados por cualquier miembro.

En la Fig. 3 se muestra como la autoridad central recolectó información (ratings) del desempeño de interacciones del pasado entre los agentes de una comunidad, y de manera continua actualizó la calificación (score) de reputación de cada agente. Con estas calificaciones actualizadas y publicadas por la autoridad central, un agente podrá tener una base para decidir en un futuro si interactúa o no, con un agente en particular.

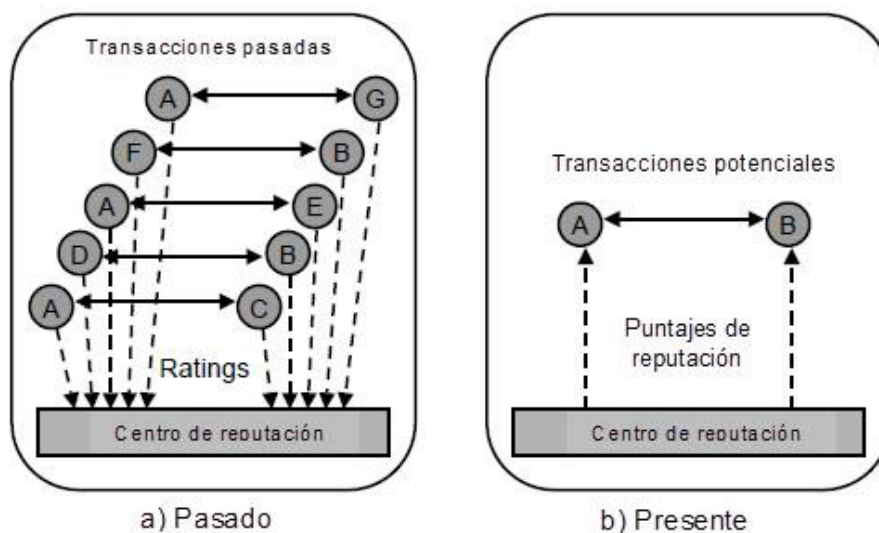


Fig. 3 Marco general de un sistema de reputación centralizado (Jøsang, et al., 2007 p 14)

Existen dos aspectos fundamentales de los sistemas de reputación centralizados:

- Protocolos de comunicación centralizados: Permiten a los participantes compartir los ratings de sus transacciones a la autoridad central, que es la misma que provee las calificaciones de reputación de los diferentes agentes para ser consultada por los participantes en futuras transacciones.

- Un motor de cálculo de reputación: Utilizado por la autoridad central para derivar la calificación de reputación de los diferentes participantes basada en los ratings recibidos (u alguna otra información).

6.1.1.2 Sistemas de Reputación Distribuidos.

Existen ambientes en los que un sistema de reputación distribuido, es más adecuado que un sistema centralizado. En un sistema distribuido no existe una locación central en donde los ratings puedan ser subidos o consultados por los participantes. En lugar de esto, el almacenamiento de la información puede estar distribuido entre los participantes, o cada participante puede compartir de manera individual sus experiencias con otros usuarios que lo requieran.

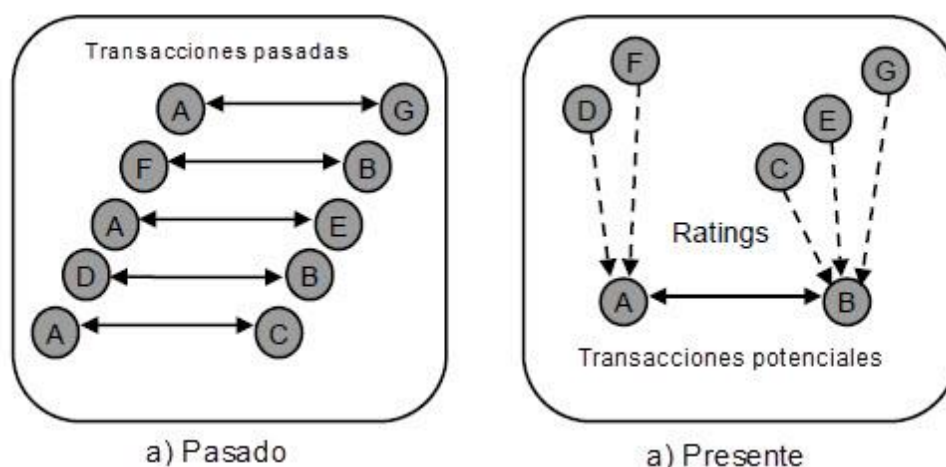


Fig. 4 Marco general de un sistema de reputación distribuido (Jøsang, et al., 2007 p 15)

Hay dos aspectos fundamentales en los sistemas de reputación distribuidos:

- Protocolos de comunicación distribuidos: Permite a los participantes obtener ratings de otros miembros de la comunidad.

- Un método de cálculo de reputación: Utilizado por cada miembro individualmente para determinar una calificación basada en los ratings recibidos u otra información.

6.1.2 Motores de cálculo de Reputación.

La calificación de reputación dentro de un sistema puede ser computada con base en experiencia propia (información privada), referencias de segunda mano (información pública) o una combinación de ambas (Jøsang et al., 2007 p. 16).

A continuación se describen algunos de los principales modelos para cálculo sistemas de Reputación (algunos ya utilizados en el área comercial y otros propuestos por la comunidad académica).

a) Sumarización Simple o Promedio de Ratings

La manera más simple de obtener calificaciones sobre reputación es a través de la sumarización de ratings positivos y negativos separadamente, para después dar una calificación total que reste los ratings positivos menos los negativos (este método es utilizado por eBay). Una de las ventajas de este método es que cualquier persona puede interpretarlas fácilmente, pero como desventaja tiene que muestra una vista muy simple.

Algunos esquemas más avanzados proponen que la calificación de reputación se obtenga a través del promedio de todos los ratings, este método se utiliza en varios sitios web (como Amazon).

Modelos más avanzados en esta categoría hacen el cálculo con promedios ponderados con base en determinado factor, como puede ser la edad del rating, un grupo de expertos que se les dé mayor peso a sus ratings, etc.

b) Sistemas Bayesianos.

Los sistemas bayesianos toman binariamente los ratings (positivos y negativos) como inputs, y están basados en el cálculo de calificaciones de reputación a través de actualizaciones estadísticas de funciones de densidad de

probabilidad. Dicha calificación *a posteriori* (actualizada) es calculada combinándola con la calificación *a priori* (previa).

La ventaja los sistemas bayesianos es que otorga teóricamente bases sólidas para el cálculo de calificaciones de reputación, pero como desventaja tiene que puede ser muy complejo para personas promedio entenderlos.

Estos sistemas expresan la probabilidad que existe de que el rating siguiente sea positivo, o con que tanta frecuencia se obtiene calificaciones positivas en los ratings.

c) Modelos Discrecionales de Confianza.

En algunos casos es más adecuado tener calificaciones en enunciados verbales de manera discrecional, que medidas continuas. Por ejemplo el tener que un agente X es Muy Confiable, Confiable, Poco Confiable, No Confiable.

d) Modelos de Creencia.

La Teoría de la Creencia es un marco de referencia relacionado a la teoría de la probabilidad, pero asume que la suma de todas las probabilidades de todas las posibles salidas no tiende a 1, y la probabilidad remanente es interpretada como incertidumbre.

e) Modelos Difusos.

Las calificaciones de reputación pueden ser representadas lingüísticamente como modelos difusos, donde funciones de pertenencia describen en qué grado puede ser descrito un agente como confiable o no confiable. La lógica difusa propone reglas para el razonamiento con medidas difusas de este tipo.

En estos modelos llaman *reputación individual* a la que se deriva de la información privada sobre un agente dado, y se llama *reputación social* a la que se deriva de la información pública de este agente, y definen como *contexto dependiente* el que deriva de la información contextual del agente.

f) Modelos de Flujo.

Sistemas que calculan la confianza o reputación por iteración transitiva a través de un ciclo (*loop*) de cadenas pueden ser llamados modelos de flujo.

En algunos modelos de flujo se asigna un peso contante de reputación/confianza al total de la comunidad, y este peso es distribuido entre los participantes de la comunidad. Los participantes solo pueden incrementar o decrementar su reputación/confianza a costo de otros participantes de la comunidad.

6.1.3 Retos en los Sistemas de Reputación

Existen numerosos problemas en los sistemas de reputación tanto los académicos como los prácticos. A continuación se mencionan los problemas más comunes:

a) Bajos incentivos al proveer un rating.

Los Ratings típicamente se dan por parte del usuario una vez después de que la transacción fue realizada, y normalmente las partes que intervienen en la transacción no tienen algún incentivo por realizar la retro con sus ratings.

Algunas compañías han optado por ofrecer recompensas monetarias o alguna recompensa a los usuarios que otorguen retroalimentación.

b) Sesgos hacia una valoración positiva.

De manera común existe un sesgo hacia valoraciones positivas. Una investigación de Resnick & Zeckhauser (2002), sobre las valoraciones de compradores y vendedores en eBay menciona que sólo un 0.6 de las valoraciones hechas por los compradores y un 1.6% de las valoraciones hechas por los vendedores son negativas, lo cual parece que no refleja la realidad. Esto se debe posiblemente a un intercambio de cortesía donde al otorgar un rating positivo se espera que la otra parte también califique positivamente, y viceversa se evita dar un rating negativo por miedo a que la otra parte devuelva una calificación negativa.

Una de las maneras en que es posible evitar este sesgo hacia ratings positivos, es a través de ratings o *reviews* se hagan de manera anónima así se evita el miedo a una posible represalia.

c) Ratings Falsos o Desleales.

Consiste en otorgar valoraciones que no reflejan la genuina opinión de la persona que los hace. Este comportamiento puede ser considerado como antiético y afecta a los miembros de la comunidad (Jøsang & Golbeck, 2009). El problema radica en que no es tan sencillo determinar cuáles valoraciones son falsas y cuales reales. Los métodos para evitar sesgos por ratings falsos pueden ser agrupados en dos categorías:

- Descuento endógeno de valoraciones desleales: Esta categoría abarca los métodos que excluyen o le dan poco peso a los ratings que presumiblemente pueden ser desleales basados en un análisis y comparación de los mismos valores. Se asume que los ratings falsos pueden ser reconocidos a través de propiedades estadísticas.
- Descuento exógeno de valoraciones desleales: Esta categoría abarca los métodos que determinan externamente la reputación de los usuarios (*raters*) que dan sus ratings, para ponderar el peso de sus valoraciones. Se asume que los usuarios con baja reputación es más probable que den ratings falsos y viceversa.

d) Cambio de Identidades.

Los sistemas de reputación presuponen que los agentes sobre los cuales se está construyendo su calificación de reputación van a tener una vida larga, que permitirá hacer transacciones en el futuro con este agente, por lo cual es relevante llevar su calificación. El cambio de identidad rompe con este supuesto, dado a que si alguna de las partes sufre alguna pérdida significativa de puntaje de reputación podría estar interesado en cambiar de identidad o un seudónimo y así cortar con el pasado y comenzar de ceros su puntaje.

e) Variaciones de Calidad a través del tiempo.

Consiste en que la calidad de algún producto o servicio puede variar a lo largo del tiempo, por distintos factores, y estos cambios en calidad necesariamente afectan su reputación. Pero esta modificación en su reputación puede ser muy lenta si ya se tienen un historial muy largo de valoraciones (por lo que no mostraría la realidad actual de un producto o servicio), por lo que uno de los métodos más utilizados para controlar esto es a través de diferentes funciones para descontar el pasado.

f) Discriminación.

Un comportamiento discriminatorio puede ocurrir al proveer un servicio o cuando se provee ratings. Por ejemplo un vendedor que siempre otorga buena calidad a todos sus compradores, excepto a un solo comprador o un grupo de compradores, por alguna causa o factor.

g) Ataque Sybil.

Una entidad establece múltiples identidades falsas para a través de una gran cantidad de valoraciones favorecer a la calificación de reputación de una entidad.

6.2 INGENIERÍA DE SOFTWARE.

Pressman, Roger (2010, p. 3) define software como “instrucciones (programas de cómputo) que cuando se ejecutan proporcionan las características, función y desempeño buscados; estructuras de datos que permiten que los programas manipulen en forma adecuada la información, e información descriptiva tanto en papel como en formas virtuales que describen la operación y uso de los mismos”.

6.2.1 Dominios de aplicación del Software.

Pressman (2010, p. 6) propone siete grandes categorías de software computacional:

- Software de Sistemas: Conjunto de programas escritos para dar servicios a otros programas. Se caracteriza por una gran interacción con el hardware de la computadora, uso intensivo por parte de usuarios múltiples, operación

concurrente que requiere la secuenciación, recursos compartidos y administración de un proceso sofisticado, estructuras complejas de datos e interfaces externas múltiples.

- Software de aplicación: Programas aislados que resuelven una necesidad específica de negocio. Las aplicaciones en esta área procesan datos comerciales o técnicos que facilitan los procesos de negocios o la toma de decisiones administrativas/técnicas.
- Software de ingeniería y ciencias: Se caracteriza por la utilización de algoritmos de grandes cálculos numéricos para el estudio o monitoreo de algún fenómeno.
- Software incrustado (embebido): Reside dentro de un producto o sistema y se utiliza para implementar y controlar funciones o características para el usuario final y el sistema en sí. El Software incrustado ejecuta funciones limitadas y particulares o provee una capacidad significativa de funcionamiento y control.
- Software de línea de productos: es diseñado para proporcionar una capacidad específica para uso de muchos consumidores diferentes. El software de línea de productos se centra en algún mercado limitado y particular (por ejemplo, control de inventario de productos) o se dirige a mercados masivos de consumidores (procesamiento de textos, hojas de cálculo, graficas por computadora, multimedios, entretenimiento, administración de base de datos, etc).
- Aplicaciones Web: Esta categoría de software centrado en redes agrupa una gama amplia de aplicaciones. Las *webapps* son poco más que un conjunto de archivos de hipertexto vinculados que presentan información con uso de texto y graficas limitadas, estas apps están evolucionando hacia ambientes de cómputo sofisticados, que se integran con bases de datos y aplicaciones de negocios.
- Software de inteligencia artificial: Utiliza algoritmos no numéricos para resolver problemas complejos que no son fáciles de tratar

computacionalmente o con el análisis directo. Las aplicaciones en esta área incluyen robótica, sistemas expertos, reconocimiento de patrones (imagen y voz), redes neuronales artificiales, demostración de teoremas y juegos.

6.2.2 Ingeniería de software.

Pressman (2010, p. 11) cita una de las primeras definiciones de ingeniería del software dada Fritz Bauer en 1969, que la define como el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficacia en máquinas reales.

El IEEE (1990 p. 67) amplía esta definición, al ver a la ingeniería de software como: 1) la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software 2) El estudio de enfoques según el punto 1.

La ingeniería de software es una tecnología con varias capas (Fig. 5). La administración total de la calidad, Six Sigma y otras filosofías similares alimentan la cultura de la mejora continua, y es esta la cultura la que lleva en última instancia al desarrollo de enfoques cada vez más eficaces de la ingeniería de software (compromiso con la calidad).



Fig. 5 Capas de la ingeniería de software (Pressman, 2010)

El fundamento para la ingeniería de software es la capa de “proceso”. Él es el aglutinante que une las capas de tecnología y permite el desarrollo racional y oportuno del software de cómputo. El proceso define una estructura que debe establecerse para la obtención eficaz de tecnología y forma la base para el control

de la administración de proyectos de software, establece el contexto en el que se aplican métodos técnicos, se generan productos del trabajo (modelos, documentos, datos, reportes, formatos, etc.), se establecen puntos de referencia, se asegura la calidad y se administra el cambio de manera apropiada.

Los métodos de la ingeniería de software proporcionan la experiencia técnica para elaborar software. Incluyen un conjunto amplio de tareas, como comunicación, análisis de los requerimientos, modelación del diseño, construcción del programa, pruebas y apoyo.

Las herramientas de la ingeniería de software proporcionan un apoyo automatizado para el proceso y los métodos. Cuando se integran las herramientas del modo que la información creada por una pueda ser utilizada por otra, se genera un sistema conocido como ingeniería de software asistido por computadora que apoya el desarrollo de software.

[2.2.1 Proceso de la Ingeniería de software.](#)

Un proceso del software es un conjunto de actividades y resultados asociados que producen un producto de software. Existen cuatro actividades fundamentales de procesos (Sommerville, 2005 p. 24):

1. Especificación del software: Los clientes e ingenieros definen el software a producir y las restricciones sobre su operación.
2. Desarrollo del software: El software se diseña y se programa.
3. Validación del software: En esta etapa se asegura que el software hace realmente lo que el cliente requiere.
4. Evolución del software: El software se modifica para adaptarlo a los cambios requeridos por el cliente y el mercado.

[2.2.1.1 Modelo de proceso del software.](#)

Un modelo de procesos del software es una descripción simplificada de un flujo del software que presenta una visión de éste. Estos modelos incluyen actividades que son parte de los procesos y productos de software y el papel de las personas involucradas en la ingeniería del software (Sommerville, 2005).

Se pueden estructurar en cinco actividades, que son aplicables a todos los proyectos de software, estas actividades son:

- Comunicación: Antes de iniciar con cualquier actividad técnica, es fundamental comunicarse y participar con el cliente. Aquí se busca la necesidad del cliente y recopilar los requerimientos que definirán las características y funciones del software.
- Planeación: Se propone el “mapa” de trabajo de los ingenieros de software al describir las tareas técnicas por realizar, los posibles riesgos, los recursos requeridos, los productos que se obtendrán y una planificación de las actividades.
- Modelado: Se crean modelos (representaciones gráficas de una realidad) con el fin de entender mejor los requerimientos y el diseño que los pueda cumplir.
- Construcción: Combina la generación de código y las pruebas que se requieran para descubrir errores en éste.
- Despliegue: El producto terminado de Software se entrega al consumidor, que lo evalúa y da retroalimentación.

La mayoría de los modelos de procesos del software están basados en uno de los tres modelos generales de desarrollo de software:

a) Modelo de cascada

El modelo de cascada: Sugiere un enfoque sistemático y secuencial para el desarrollo de software. Considera las actividades anteriores y las representa como fases de procesos separados, después de que cada etapa se termina, el desarrollo continúa con la siguiente etapa (Fig. 6).

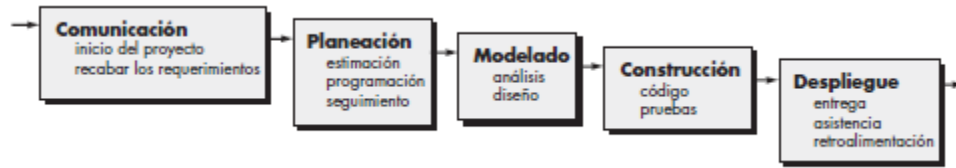


Fig. 6 Modelo de cascada (Pressman, 2010)

Este modelo es de los más antiguos en el desarrollo de software, sin embargo actualmente los ingenieros de software prefieren utilizar otro tipo de paradigma ya que en la realidad es raro que los proyectos sigan un flujo secuencial, durante el proyecto se pueden presentar cambios en los requerimientos y también el cliente no puede realizar pruebas hasta que ya está muy avanzado el proyecto, lo que puede ser muy costoso si se desea hacer algún cambio.

b) Desarrollo iterativo

Un sistema inicial se desarrolla rápidamente a partir de especificaciones muy abstractas. Éste se refina basándose en las peticiones del cliente para producir un sistema que satisfaga las necesidades de dicho cliente. El sistema hasta entonces puede ser entregado.

Hay dos modelos comunes de procesos iterativos:

- Haciendo prototipos: Se crea una primera versión de software de manera rápida a partir de objetivos generales, llevando a cabo todas las actividades del proceso, y al final se evalúa el producto y se da una retroalimentación, esta información se aplica para mejorar el prototipo en una segunda versión (ver fig. 7).

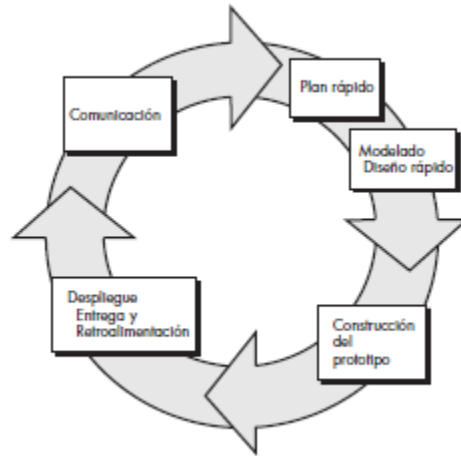


Fig. 7 Modelo Iterativo de prototipos (Pressman, 2010 p 37)

- **Modelo espiral:** Es un modelo evolutivo que se acopla con la naturaleza iterativa de hacer prototipos con los aspectos controlados y sistémicos del modelo de cascada. Se hace el desarrollo de versiones de manera cada vez más completa.

El modelo de desarrollo espiral tiene dos características distintivas. La primera es el enfoque cíclico para el crecimiento incremental del grado del grado de definición de un sistema y su implementación, mientras que disminuye el grado de riesgo. La otra es un conjunto de puntos de referencia de anclaje puntual para asegurar el compromiso del participante con soluciones factibles y mutuamente satisfactorias.

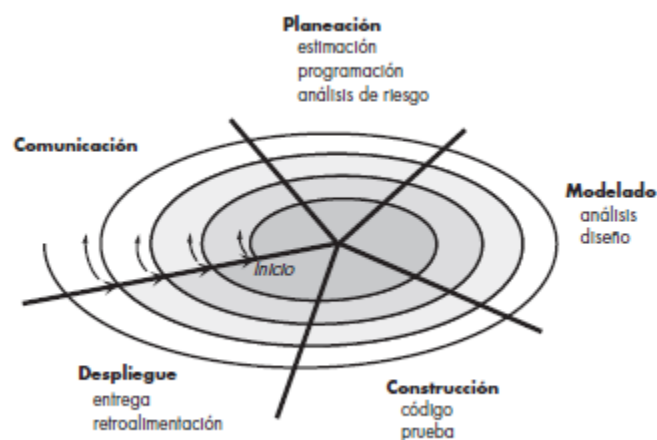


Fig. 8 Modelo Iterativo de espiral (Pressman, 2010 p 37)

c) Ingeniería del software basada en componentes (CBSE):

Esta técnica supone que las partes del sistema existen. El proceso de desarrollo del sistema se enfoca en la integración de estas partes desarrolladas desde el principio. Las actividades de modelado y construcción comienzan con la identificación de candidatos de componentes. Éstos se pueden diseñar como módulos de software o clases orientadas objetos o paquetes.

El modelo del desarrollo basado en componentes lleva la reutilización del software, lo cual da a los ingenieros de software varios beneficios.

2.2.3 Métodos de la ingeniería del software.

Un método de ingeniería del software es un enfoque estructurado para el desarrollo de software cuyo propósito es facilitar la producción de software de alta calidad de una forma costeable.

No existe un método ideal, y métodos diferentes tienen distintas áreas donde son aplicables. Todos los métodos se basan en la idea de modelos gráficos de desarrollo de un sistema y en el uso de estos modelos como un sistema de especificación o diseño. Los métodos incluyen varios componentes:

Tabla 3. Componentes del método

Componentes del método	
Descripciones del modelo del sistema	Descripciones de los modelos del sistema que desarrollará la notación utilizada para definir los modelos.
Reglas	Restricciones que siempre aplican a los modelos de sistemas.
Recomendaciones	Heurística que caracteriza una buena práctica de diseño en este método. Seguir estas recomendaciones debe dar como resultado un modelo del sistema bien organizado.

Guías en el proceso	Descripciones de las actividades que deben seguirse para desarrollar los modelos del sistema y la organización de estas actividades.
---------------------	--

Fuente: Sommerville, 2005

Tabla 4. Atributos del método

Atributos del método	
Mantenibilidad	El software debe de escribirse de tal forma que pueda evolucionar para cumplir necesidades de cambio de los clientes. Éste es un atributo crítico debido a que el cambio en el software es una consecuencia inevitable de un cambio en el entorno de los negocios.
Confiabilidad	La confiabilidad del software tiene un gran número de características, incluyendo la fiabilidad, protección y seguridad. El software confiable no debe causar daños físicos o económicos en el caso de una falla del sistema.
Eficiencia	El software no debe de hacer que se malgasten los recursos del sistema, como la memoria y los ciclos de procesamiento.
Usabilidad	El software debe de ser fácil de utilizar, sin esfuerzo adicional, por el usuario para quien está diseñado. Esto significa que debe de tener una interfaz de usuario apropiada y una documentación adecuada.

Fuente: Sommerville, 2005

6.3 APLICACIONES MÓVILES.

Una aplicación móvil (*app*) es una aplicación de software específicamente desarrollada para utilizarse en dispositivos móviles. Estas aplicaciones deben de

considerar en su diseño las características de los aparatos en que van a correr, ya que cuentan con limitaciones de recursos (almacenamiento, procesamiento o memoria), retos de conectividad, distintas entradas de datos y una gran variedad de resoluciones de pantalla (Nayebi, et al.2012).

Las aplicaciones móviles han ido evolucionando desde los primeros aparatos móviles, en un inicio estas apps estaban diseñadas para ayudar al usuario en un tema en específico (calendario, agenda de contactos, pequeños editores de texto, y más adelante, clientes de correo y navegadores web), pero con el paso del tiempo al hacerse más barato los componentes de los dispositivos móviles estos fueron ganando en recursos y capacidades, lo que permitió que estas aplicaciones móviles hicieran cada vez más cosas.

Las apps móviles se pueden dividir en tres tipos (Cuello & Vittone, 2013):

- **Aplicaciones nativas:** son aquellas que han sido desarrolladas con el software que ofrece cada sistema operativo a los programadores, llamado genéricamente *Software Development Kit* (SDK). Así, Android, iOS y Windows Phone tienen uno diferente y las aplicaciones nativas se diseñan y programan específicamente para cada plataforma, en el lenguaje utilizado por el SDK.

Este tipo de apps se descarga e instala desde las tiendas de aplicaciones, aprovechando de las diferentes herramientas de promoción y marketing de cada una de ellas.

Las aplicaciones nativas se actualizan frecuentemente, el usuario debe descargarlas de nuevo para obtener la última versión, que corrige o añade mejoras con respecto a la versión previa

Las aplicaciones nativas pueden hacer uso de las notificaciones del sistema operativo para mostrar avisos importantes al usuario, aun cuando la aplicación no esté en uso (pueden correr en *background*).

No requieren Internet para funcionar, por lo que ofrecen una experiencia de uso más fluida ya que están realmente integradas al teléfono, lo cual les

permite utilizar todas las características de hardware del terminal, como la cámara y los sensores (GPS, acelerómetro, giróscopo, etc) (Prieto Blázquez, et al., 2011 p. 103).

A nivel de diseño, esta clase de aplicaciones tiene una interfaz basada en las guías de cada sistema operativo, logrando mayor coherencia y consistencia con el resto de aplicaciones y con el propio SO. Esto favorece la usabilidad y beneficia directamente al usuario que encuentra interfaces familiares.

- **Aplicaciones web:** La base de programación de las aplicaciones web (también llamadas webapps) es el HTML, conjuntamente con JavaScript y CSS, herramientas ya conocidas para los programadores web.

En este caso no se emplea un SDK, lo cual permite programar de forma independiente al sistema operativo en el cual se usará la aplicación. Por eso, estas aplicaciones pueden ser fácilmente utilizadas en diferentes plataformas sin mayores inconvenientes y sin necesidad de desarrollar un código diferente para cada caso particular.

Las aplicaciones web no necesitan instalarse, ya que se visualizan usando el navegador del teléfono como un sitio web normal. Por esta misma razón, no se distribuyen en una tienda de aplicaciones, sino que se comercializan y promocionan de forma independiente.

Al tratarse de aplicaciones que funcionan sobre la web, no es necesario que el usuario reciba actualizaciones, ya que siempre va a estar viendo la última versión. Pero, a diferencia de las apps nativas, requieren de una conexión a Internet para funcionar correctamente (Prieto Blázquez et al., 2011).

Adicionalmente, tienen algunas restricciones e inconvenientes en factores importantes como gestión de memoria y no permiten aprovechar al máximo la potencia de los diferentes componentes de hardware del teléfono.

Las aplicaciones web suelen tener una interfaz más genérica e independiente de la apariencia del sistema operativo, por lo que la

experiencia de identificación del usuario con los elementos de navegación e interacción, suele ser menor que en el caso de las nativas.

- **Aplicaciones híbridas:** Este tipo de aplicaciones es una combinación entre las dos anteriores. La forma de desarrollarlas es parecida a la de una aplicación web (usando HTML, CSS y JavaScript), y una vez que la aplicación está terminada, se compila o empaqueta de forma tal, que el resultado final es como si se tratara de una aplicación nativa.

Esto permite casi con un mismo código obtener diferentes aplicaciones, por ejemplo, para Android y iOS, y distribuirlas en cada una de sus tiendas.

A diferencia de las aplicaciones web, estas permiten acceder, usando librerías, a las capacidades del teléfono, tal como lo haría una app nativa.

Las aplicaciones híbridas, también tienen un diseño visual que no se identifica en gran medida con el del sistema operativo. Sin embargo, se pueden usar controles y botones nativos de cada plataforma para apegarse más a la estética propia de cada una.

6.3.1 Métodos aplicados al desarrollo de aplicaciones móviles

Como ya se mencionó en el apartado de ingeniería de software existen ya métodos utilizados para el desarrollo de software, y en el caso de desarrollo de aplicaciones móviles, se utilizan también estos métodos.

Una de las características importantes de la gran mayoría de los desarrollos móviles es su corta duración. Esto se debe a factores como la gran competencia en el sector, los cambios en el mismo con la aparición, casi constante, de novedades tanto software como hardware, el hecho de que muchas aplicaciones nacen con un desarrollo precoz en forma de prototipo (y van evolucionado después) o incluso la simplicidad de las aplicaciones, que no requieren grandes desarrollos. Esta suele ser, salvo algunas excepciones, la norma de los desarrollos de aplicaciones para dispositivos móviles (Prieto Blázquez, et al., 2011).

6.3.1.1 Modelo de cascada

Como se menciona en la sección de ingeniería de software el modelo *cascada* es el modelo más estático y predictivo. Para móviles es aplicable en proyectos en los que los requisitos están fijados y no van a cambiar durante el ciclo de vida del desarrollo, proyectos realmente controlados y previsibles, en los que no hay mucha incertidumbre por lo que se desea hacer y para los que no son importantes los cambios constantes en la industria.

6.3.1.2 Desarrollo rápido de aplicaciones

Éste es un método iterativo cuyo objetivo es conseguir prototipos lo antes posible para mejorarlos después, poco a poco. Se suele priorizar la implementación sobre la planificación, y se utilizan muchos patrones de diseño conocidos para poder adaptarse de la mejor manera a cambios en los requerimientos.

El desarrollo rápido de aplicaciones es un método muy útil para el desarrollo de proyectos realmente urgentes con tiempos de entrega muy cortos.

6.3.1.3 Desarrollo ágil

El desarrollo ágil está basado en iteraciones, donde en cada iteración se realizan todas las fases del ciclo de desarrollo.

El manifiesto ágil fue publicado en el 2001 por diecisiete desarrolladores de software, quienes representaban entonces los métodos de desarrollo más populares, que pasarían a conocerse como ágiles (*Extreme Programming*, *Crystal Clear*, *DSDM* o *ASD*, entre otros). El manifiesto define doce principios y cuatro valores éticos para los desarrolladores (Beck et al., 2001).

El desarrollo ágil se basa en los principios del manifiesto ágil y sus valores éticos, que tratan de dar más valor a algunos conceptos, pero sin dejar de lado los demás. Son los siguientes:

- 1) Dar más valor a los individuos y a sus interacciones que a los procesos y herramientas.

- 2) Dar más valor al software que funciona que a la documentación exhaustiva.
- 3) Dar más valor a la colaboración con el cliente que a la negociación contractual.
- 4) Dar más valor a la respuesta al cambio que al seguimiento de un plan.

Con estos valores se intenta conseguir, entre otras cosas, entregar algo lo más pronto posible y evitar problemas originados por cambios de requisitos. Esto es muy apropiado para proyectos cambiantes, ya sean grandes o pequeños, ya que mediante estos valores se pueden mitigar los riesgos. Para conseguir proyectos que puedan cambiar fácilmente, se pone especial atención en la calidad de los productos conseguidos, cosa que es realmente importante en proyectos de software para dispositivos móviles. Para conseguir esto, se basan en las pruebas de la aplicación y, a menudo, las automatizan.

Los métodos ágiles suelen ser muy adecuados para el desarrollo de aplicaciones móviles por las siguientes razones:

- Alta volatilidad del entorno: Con cambios en entornos de desarrollo, nuevos terminales y nuevas tecnologías a un ritmo mucho más elevado que en otros entornos de desarrollo.
- Equipos de desarrollo pequeños: Dado que los desarrollos móviles suelen ser proyectos relativamente pequeños, los equipos no suelen ser muy grandes. Generalmente son llevados a cabo por desarrolladores individuales o por PYME.
- Software no crítico: No suelen ser aplicaciones de alto nivel de criticidad, dado que suelen ser aplicaciones para entretenimiento o gestión empresarial no crítica.
- Ciclos de desarrollo cortos: Dada la evolución constante de la industria, se requieren ciclos de vida realmente cortos para poder dar salida a las aplicaciones a tiempo.

6.3.1.4 Mobile-D

El método Mobile-D se desarrolló junto con un proyecto finlandés en el 2004. Fue realizado, principalmente, por investigadores de la VTT (Instituto de Investigación Finlandés) y, a pesar de que es un método antiguo, sigue en vigor (se está utilizando en proyectos de éxito y está basado en técnicas que funcionan).

El objetivo es conseguir ciclos de desarrollos muy rápidos en equipos muy pequeños (de no más de diez desarrolladores) trabajando en un mismo espacio físico. Según este método, trabajando de esa manera se deben conseguir productos totalmente funcionales en menos de diez semanas (Ramírez Vique, 2013).

Se trata de método basado en soluciones conocidas y consolidadas: *Extreme Programming* (XP), *Crystal Methodologies* y *Rational Unified Process* (RUP), XP para las prácticas de desarrollo, Crystal para escalar los métodos y RUP como base en el diseño del ciclo de vida.

Cada fase (excepto la inicial) tiene siempre un día de planificación y otro de entrega. Las fases son (ver Fig. 9):

- Exploración. Se dedica a la planificación y a los conceptos básicos del proyecto. Es diferente del resto de fases.
Inicialización. Se preparan e identifican todos los recursos necesarios. Se establece el entorno técnico.
- Fase de producto. Se repiten iterativamente las subfases, con un día de planificación, uno de trabajo y uno de entrega. Aquí se intentan utilizar técnicas como la del *test driven development* para conseguir la mayor calidad.
- Fase de estabilización. Se llevan a cabo las acciones de integración para asegurar que el sistema completo funciona correctamente.
- Fase de pruebas y reparación. Tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema según los requisitos del cliente.

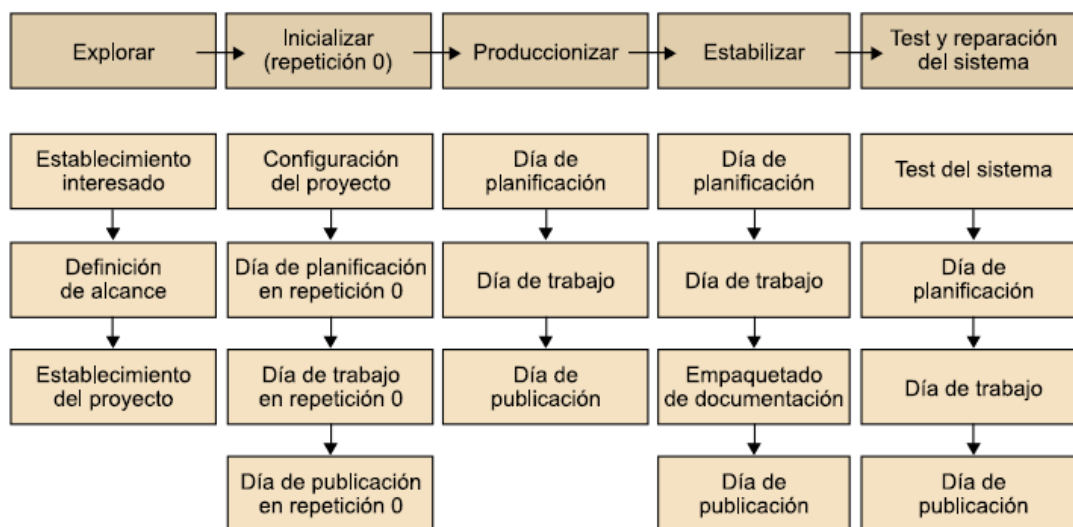


Fig. 9 Ciclo de desarrollo de Mobile-D (Ramírez Vique, 2013 p 42)

6.3.2 Categorías de aplicaciones móviles:

Una manera de agrupar las aplicaciones (*apps*) se basa en el tipo contenido que ofrecen al usuario. En base a la categoría se toman decisiones del diseño de la interfaz de la app y la manera en monetizará.

Aunque hay ocasiones en que una app entra en más de una categoría, estas son las cinco principales categoría de apps, con base en el contenido que ofrecen (Cuello & Vittone, 2013):

a) Entretenimiento

Estas son las apps de juegos y aquellas que de una forma u otra, proponen diversión para el usuario. Gráficos, animaciones y efectos de sonido intentan mantener la atención constante e ininterrumpida en lo que está sucediendo en la pantalla. Generalmente, presentan un diseño no tan apegado a las pautas de su plataforma.

En cuanto al modelo de negocio, son flexibles porque pueden descargarse pagando por versiones completas u ofrecer otras posibilidades de compra: por ítems, por niveles, etc.

b) Sociales

Son aquellas que se orientan principalmente a la comunicación entre personas, creación de redes de contactos e interacción entre usuarios.

Normalmente son gratuitas y su modelo de negocio está en la información personal que se obtiene de los usuarios o en las compras dentro de la app.

c) Utilitarias y productividad

Mayormente asociadas con el sector empresarial, estas aplicaciones proporcionan herramientas para solucionar problemas específicos y se basan en la ejecución de tareas concretas, cortas y rápidas. Se privilegia la eficiencia sobre todo lo demás.

En esta clasificación se encuentran comúnmente las listas de tareas o aquellas apps orientadas a equipos de trabajo, donde los usuarios dan más valor a aquellas herramientas que permitan simplificar sus tareas diarias.

En esta categoría el modelo de negocio es variable. Si la aplicación se encuentra solo disponible para móvil, lo normal es que se pague por la descarga. En cambio, las que están asociadas a un servicio en la nube por el cual ya se paga, se descargan de forma gratuita.

d) Educativas e Informativas

Las aplicaciones educativas y de información se usan como transmisores de conocimiento y noticias. En estas apps se privilegia el acceso al contenido, por este motivo, la legibilidad, facilidad de navegación y herramientas de búsqueda son fundamentales.

e) Creación

Estas apps ponen el foco en la creatividad del usuario y en ofrecerle herramientas para potenciarla. Por ejemplo, aquellas que permiten editar vídeos, retocar fotografías, producir sonidos o escribir.

Aunque suelen ser de pago, algunas ofrecen versiones gratuitas (no tan completas) o añaden componentes extra que deben pagarse individualmente.

6.3.3 Monetización de apps

Las apps pueden ser gratis o de pago, esto va a depender de la intención del fabricante y el objetivo de la app, puede ser que el objetivo de la app no sea conseguir dinero, sino servir como canal de comunicación con los usuarios o como una forma de extender el alcance de marca (Cuello & Vittone, 2013). De cualquier forma, se trata de una decisión personal (en el caso de los desarrolladores independientes) o que responde a objetivos globales (para las marcas).

Apps gratuitas

El mayor beneficio que se puede obtener de una aplicación gratuita es el alcance o la cantidad de usuarios potenciales a los que puede llegar, ya que no hay ninguna barrera de entrada para que un usuario descargue la aplicación y empiece a probarla (Cuello & Vittone, 2013).

Este primer paso es fundamental para el conocimiento de la aplicación: quien la descargue no tiene nada que perder. Esto disminuye en cierta forma las expectativas sobre el producto y puede servir para ciertos casos en los que el producto aún tiene camino de desarrollo por delante hasta su maduración. En este caso, su colocación gratuita en el mercado, permite usar ciertos indicadores del comportamiento de los usuarios (cómo usan la app o qué tan frecuentemente, por ejemplo) para mejorar futuras versiones.

Un inconveniente de las aplicaciones gratuitas es que, debido a la gran competencia, es más difícil obtener visibilidad en los rankings de mejores aplicaciones de las diferentes tiendas. Además, requieren una cantidad total de descargas mayor que la necesaria en las aplicaciones de pago para llegar a los primeros puestos.

Una aplicación gratuita también puede servir como ventana para promocionar una versión de pago de ella misma u otras aplicaciones del mismo desarrollador.

Que una app sea gratuita, no quiere decir que no se pueda obtener ningún dinero con ella, ya que existen estrategias de monetización ya estando dentro de la app.

Apps de pago

Es más complicado que una aplicación de pago sea exitosa porque requiere un gran número de descargas para ser rentable; además, poner un precio a la descarga suele ser una barrera difícil para el usuario que no se quiere arriesgar a pagar por algo que aún no conoce.

Que un usuario esté dispuesto a pagar o no por una app, puede depender de varios factores. Uno tiene que ver con las alternativas gratuitas que existan. Si dos aplicaciones son similares (a nivel de funcionalidad y diseño), pero una de ellas es de pago y la otra no, lo más probable es que el usuario descargue la aplicación gratuita (Cuello & Vittone, 2013).

La tienda donde se encuentre la app también influye en las posibilidades de cobrar por ella. Por ejemplo, Los usuarios de iPhone están más acostumbrados a pagar por una aplicación que alguien que usa Android o Windows Phone; pero, quizás por esta misma razón, demanda más calidad para cubrir sus expectativas.

Independientemente de la plataforma donde la app se encuentre, el usuario paga por el valor, por algo que la app le aporte y que justifique su precio, un precio que muchas veces, está condicionado por el mercado y por la competencia.

Un parámetro importante que influye en los usuarios antes de pagar por una app es su valoración. Cuantas más y más altas valoraciones tenga, es más probable que un usuario pague por una app con muchas calificaciones y una media de valoraciones altas, que por una que no tenga ninguna valoración o que tenga valoraciones negativas.

Las distintas tiendas (tanto Google Play, como App Store o Windows Phone Store) cobran una comisión de 30% sobre el precio de venta de la app. El desarrollador se queda solo con el 70% de lo que el usuario pague por su descarga.

Freemium

Este modelo *freemium* es una combinación de los dos anteriores. Su nombre viene de la mezcla de las palabras inglesas *free* y *premium*, y consiste en descargar la aplicación de forma gratuita, permitiendo al usuario un uso básico y limitado, con la posibilidad de recibir funciones más avanzadas, pagando por ellas.

Combina la fortaleza de ambos modelos: la app puede llegar a un número mayor de personas al ser gratuita, pero también permite ofrecer servicios avanzados a los usuarios que realmente la encuentran útil o quieren sacarle el máximo provecho.

La dificultad reside en determinar qué partes de la aplicación ofrecer de manera gratuita y por cuáles es conveniente cobrar. Otorgar demasiadas funciones gratuitas hará que poca gente quiera adquirir las de paga, pues la versión gratuita les resultará suficiente. Por otro lado, si la mayoría de funciones son de pago, los usuarios pueden encontrar la aplicación poco práctica y dejarán de usarla.

Un ejemplo son los juegos que dejan avanzar ciertos niveles, pero para llegar hasta el final hay que pagar por la versión completa.

6.3.3.1 Modelos de monetización

“Los modelos de monetización son diferentes caminos para obtener dinero a través de las aplicaciones. No deben verse de forma individual y separada, ya que suelen depender de si la app es gratis, paga o freemium, y también de la categoría de la aplicación” (Cuello & Vittone, 2013 p. 39).

a) Compras dentro de la app (In-app purchase)

Algunas apps permiten pagar pequeñas cantidades de dinero por la compra de ítems separados que mejoran las prestaciones básicas; por ejemplo, algunas aplicaciones de fotografía venden filtros de fotos avanzados. Es también el caso de las apps que ofrecen contenidos *premium* o suscripciones. Esta forma de monetización está más asociada a la distribución *freemium*.

b) Pagar por la versión completa

En este caso, las aplicaciones se desarrollan en dos versiones: una gratuita con funciones básicas, que permite al usuario probarla con algunas limitaciones o publicidad y una de pago, que ofrece más posibilidades para quien esté convencido de la compra luego de haber probado la versión gratuita.

Este modelo está usándose cada vez menos por los inconvenientes que presenta tener dos aplicaciones separadas. Por un lado, los usuarios que quieren la versión *premium* tienen que descargar una nueva aplicación, con la dificultad que esto ocasiona, pues a nivel de desarrollo no siempre es fácil trasladar la configuración que ha definido el usuario de una versión a la otra.

Además, las valoraciones de los usuarios y la posición dentro de los rankings de las tiendas de aplicaciones son independientes para cada una de las versiones.

No obstante, estos inconvenientes pueden evitarse ofreciendo la versión *premium* como una compra dentro de la app.

c) Publicidad

La publicidad puede usarse en aplicaciones gratuitas como herramienta para obtener rédito económico. Suele presentarse en forma de pequeños avisos que pueden ser pulsados por el usuario para acceder a otras webs o descargar otras aplicaciones. En este modelo la ganancia depende de la cantidad de gente que entre a los anuncios.

Como principales inconvenientes se pueden nombrar la intrusión a la privacidad del usuario y que la visualización de avisos afecta la experiencia general.

Cada sistema operativo tiene su propio sistema de publicidad y en cada uno de ellos las condiciones serán diferentes. Google ofrece un programa de publicidad para las aplicaciones en Android llamado Google AdMob; por su parte, para iOS se encuentra iAd y para Windows Phone, Microsoft Advertising.

Al considerar la opción de incluir publicidad, es muy importante pensar en qué plataformas se quiere distribuir la aplicación. Los avisos pueden verse como algo normal en Android, ya que muchos de sus usuarios prefieren ver anuncios a pagar por una app, pero pueden ser un problema en aplicaciones para iOS donde los usuarios tienen un comportamiento diferente frente a la publicidad (Cuello & Vittone, 2013).

7. DESARROLLO DE LA APLICACIÓN.

Como ya se mencionó dentro del objetivo la finalidad de este proyecto es desarrollar una aplicación móvil a nivel piloto, que funcione como una plataforma colaborativa donde los usuarios (consumidores de combustible) puedan subir calificaciones de sus experiencias en las diferentes gasolineras, donde carguen combustible.

La metodología para desarrollar esta aplicación será una de tipo cascada, aunque tomará conceptos de la metodología ágil, ya que estará más enfocada al funcionamiento y pruebas funcionales que a la documentación, con las fases que se mencionan a continuación:

- Fase 1. Análisis:
 - ° Obtención y clasificación de requerimientos.
 - ° Propuesta de solución.
- Fase 2. Diseño:
 - ° Definir escenarios.
 - ° Arquitectura del software.
 - ° Diseñar/plot de la secuencia de la App.

- Fase 3: Desarrollo:
 - ° Creación de la base de datos y Servicios Web.
 - ° Creación de las pantallas y visualización grafica de la App.
 - ° Codificación.
 - ° Pruebas unitarias.
- Fase 4 Pruebas de Funcionamiento:
 - ° Emulación y simulación.
 - ° Dispositivos reales.
 - ° Pruebas integrales.
- Fase 5 Liberación/Publicación:
 - ° Publicación del APK.

7.1 ANÁLISIS.

7.1.1 Obtención y clasificación de requerimientos

a) *Objetivos específicos del proyecto*

ID	OBJETIVO	PRIORIDAD
OB01	Desarrollo de una aplicación móvil que sirva para consultar e ingresar datos sobre la reputación de las gasolineras en México	Alta
OB02	Crear una base de datos donde se guarden y se procese la información de reputación sobre las gasolineras en México.	Alta

b) *Descripción de requerimientos específicos*

- Requerimientos funcionales

Descripción de Requerimiento:	
ID de Requerimiento: RQ01	Tipo de Requerimiento: Funcional
Descripción: El usuario deberá poder calificar la experiencia que tuvo al cargar gasolina en determinada gasolinera a través de la app (Dar <i>like</i> o <i>dislike</i>).	
Prioridad: Alta	
Fuente: OB01	
Criterio de Éxito: Prototipo de la app funcionando en sistemas operativos Android 4.1 y posteriores, que al	

presionar los botones de *like* o *dislike* actualicen la base de datos en SQL.

Historial:

Descripción de Requerimiento:

ID de Requerimiento: RQ02

Tipo de Requerimiento: Funcional

Descripción: La aplicación deberá de mostrar un mapa con la ubicación e información de las gasolineras en México, mostrándolas con alertas tipo semáforo dependiendo de su calificación en tiempo real.

Prioridad: Alta

Fuente: OB01

Criterio de Éxito: Prototipo de la app funcionando en sistemas operativos Android 4.1 y posteriores, que refleje de manera correcta la información que se encuentra en la base de datos de SQL.

Historial:

Descripción de Requerimiento:

ID de Requerimiento: RQ03

Tipo de Requerimiento: Funcional

Descripción: La aplicación deberá poder mostrar la geolocalización del usuario (GPS) dentro de un mapa de Google maps.

Prioridad: Media

Fuente: OB01

Criterio de Éxito: Prototipo de la app funcionando en sistemas operativos Android 4.1 y posteriores, mostrando un punto con la ubicación geográfica del usuario, y cambiando de acuerdo al movimiento de éste.

Historial:

Descripción de Requerimiento:

ID de Requerimiento: RQ04

Tipo de Requerimiento: Funcional

Descripción: La aplicación deberá el historial de calificaciones de las estaciones de servicios.

Prioridad: Media

Fuente: OB01

Criterio de Éxito: Prototipo de la app funcionando en sistemas operativos Android 4.1 y posteriores, mostrando de manera correcta la información historia de las calificaciones de las gasolineras de acuerdo a la base de datos en el servidor.

Historial:

Descripción de Requerimiento:

ID de Requerimiento: RQ05

Tipo de Requerimiento: Funcional

Descripción: Mostrar el precio de las gasolinas oficial y por estación de servicio.

Prioridad: Media

Fuente: OB01

Criterio de Éxito: Prototipo de la app funcionando en sistemas operativos Android 4.1 y posteriores, mostrando de manera correcta la información del precio de la gasolina actualizado en cada estación de servicio.
Historial:

- Requerimientos no funcionales

Descripción de Requerimiento:	
ID de Requerimiento: RQ06	Tipo de Requerimiento: No Funcional
Descripción: La aplicación móvil se debe crear en Android Studio y deberá funcionar para el sistemas operativos Android en versión Jelly Bean y posteriores.	
Prioridad: Alta	
Fuente: OB01	
Criterio de Éxito: Prototipo de la app funcionando en sistemas operativos Android 4.1 y posteriores.	
Historial:	

Descripción de Requerimiento:	
ID de Requerimiento: RQ07	Tipo de Requerimiento: No Funcional
Descripción: Web services (.NET) publicados en un servidor (Windows Server 2008) que puedan ser invocados través de la aplicación móvil.	
Prioridad: Media	
Fuente: OB02	
Criterio de Éxito: Conexión y actualización de datos exitosamente desde la app.	
Historial:	

Descripción de Requerimiento:	
ID de Requerimiento: RQ08	Tipo de Requerimiento: No Funcional
Descripción: Base de datos SQL Express 2008, donde se alojaría la información de la reputación de las gasolineras.	
Prioridad: Alta	
Fuente: OB02	
Criterio de Éxito: Conexión y actualización de datos exitosamente desde la app.	
Historial:	

Descripción de Requerimiento:	
ID de Requerimiento: RQ09	Tipo de Requerimiento: No Funcional
Descripción: El equipo móvil donde se instale la app deberá contar con acceso a internet y GPS activo.	
Prioridad: Alta	
Fuente: OB01	
Criterio de Éxito: Prototipo de la app funcionando en sistemas operativos Android 4.1 y posteriores.	
Historial:	

Descripción de Requerimiento:	
ID de Requerimiento: RQ10	Tipo de Requerimiento: No Funcional
Descripción: La lista de gasolineras y sus ubicaciones vendrán de la información publicada por la PROFECO.	
Prioridad: Alta	
Fuente: OB01	
Criterio de Éxito: Aplicación mostrando las gasolineras en las ubicaciones adecuadas.	
Historial:	

7.1.2 Propuesta de solución

La propuesta se enfocó en realizar una aplicación a nivel piloto, que funcione como una plataforma colaborativa donde los usuarios (consumidores de combustible) puedan subir calificaciones de sus experiencias en las diferentes gasolineras, donde carguen combustible.

Esta app debe de contar con un mapa con las diferentes ubicaciones de gasolineras en México, donde el usuario pueda hacer *check-in* al cargar en alguna de estas y al final dar un *like* o *dislike* en esa gasolinera en base al servicio, si considera que lo que pago en litros es igual a lo que realmente cargó.

Esta retro de los usuarios modificara la calificación (reputación) de cada estación de servicios mostrándola dentro del mapa con colores de tipo semáforos, clasificándolas como buenas, regulares y malas.

La app deberá mostrar la ubicación del usuario con base en su GPS del teléfono y las gasolineras próximas a su ubicación.

Para esta solución será necesario tener un servidor en la nube donde esté alojada la base de datos de las calificaciones de las gasolineras, y se estaría comunicando con los diferentes dispositivos a través de *Web Services*, tanto para mandarles información actualizada como para recibir la retro de los usuarios.

También contaría con otra información relevante como la información del precio actual de los combustibles, y es posible que en un futuro entren nuevas marcas a ofrecer combustible a México, por lo que es probable que cada una tenga un precio diferente, donde ayudaría esta app al poder mostrar el precio actual en cada una de las gasolineras, y así el usuario pueda tomar una mejor decisión.

7.2 DISEÑO.

7.2.1 Definir escenarios

De acuerdo con la propuesta de solución es necesario crear escenarios de las principales funciones de la aplicación móvil. Estos escenarios incluyen los componentes que al trabajar en sincronía con la app, hacen que la propuesta de solución sea posible.

El primer escenario (inicio) muestra cómo debería de funcionar la app al iniciarla (Fig. 10), donde el usuario al abrir la app, lo primero que hará la aplicación será determinar la ubicación del usuario, con esta ubicación la app por medio de un WS hará una llamada a un servidor en la nube, donde estará alojada la información de la ubicación y la reputación de las diferentes gasolineras en México, este servidor procesará la llamada del WS y mandará la información de las gasolineras cercanas a la ubicación del usuario. La app recibirá la información del WS y con esta información pondrá marcadores dentro de un mapa (Google map) que de acuerdo a la calificación de la gasolinera será el color que le ponga al marcador (verde, amarilla o roja).

Este proceso se repetirá cada vez que se regrese a la pantalla principal.

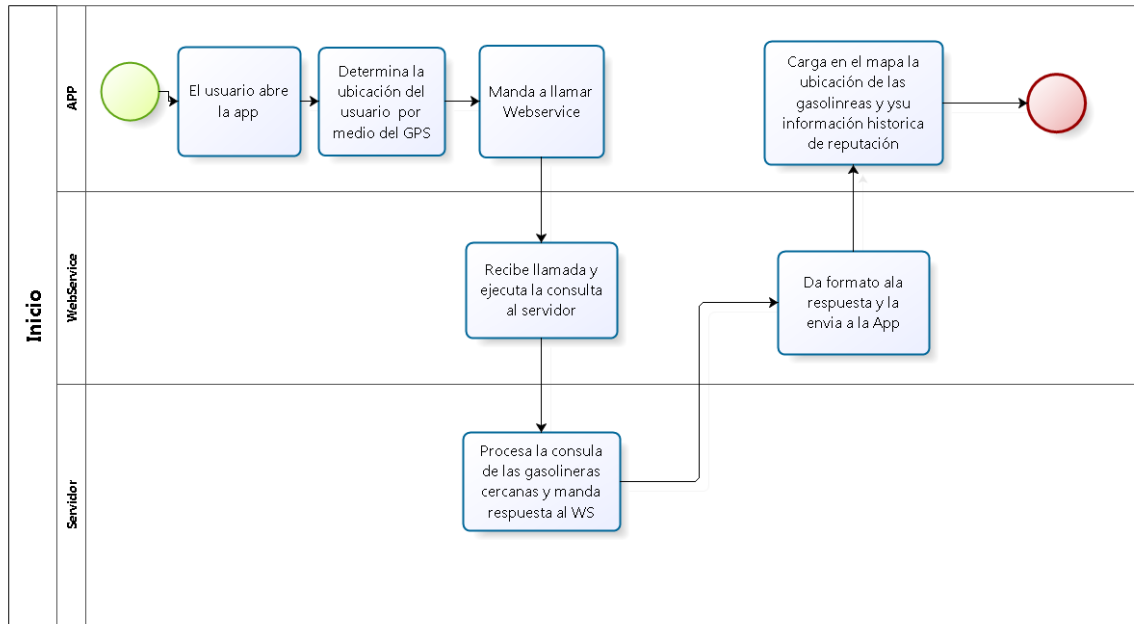


Fig. 10 Proceso del inicio de la APP (Elaboración propia).

Un segundo escenario se presentaría cuando el usuario desea ver información sobre alguna gasolinera en particular, para esto daría clic sobre la estación que le gustaría obtener más información y el resultado se le mostraría en una pantalla nueva, filtrando únicamente esta gasolinera, para lograr esto, al igual que en el escenario uno, se utilizarían WS para comunicarse con el servidor, este proceso se puede ver en la Fig. 11:

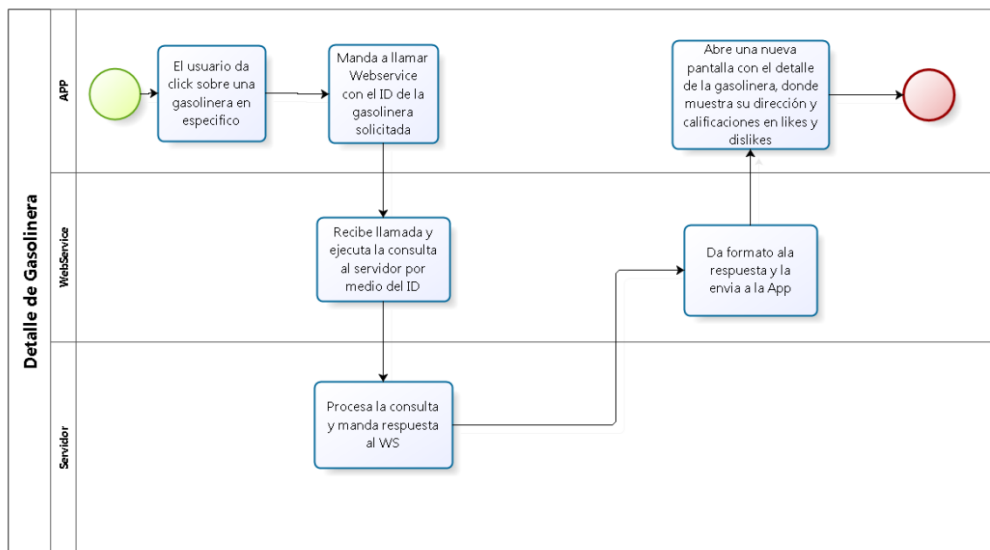


Fig. 11 Proceso del detalle de Gasolineras (Elaboración propia).

Un último escenario sería cuando el usuario califica una estación de servicio, por medio de un *like* o *dislike*, con base en su experiencia en esa gasolinera y si considera si lo despachado está en relación con el cobro que se le hizo. Para esto dentro de la pantalla de detalle de una gasolinera en específico, el usuario manda su respuesta, que se procesa en el servidor y actualiza la información de esta gasolinera en la base de datos, para después mostrar la calificación actualizada.

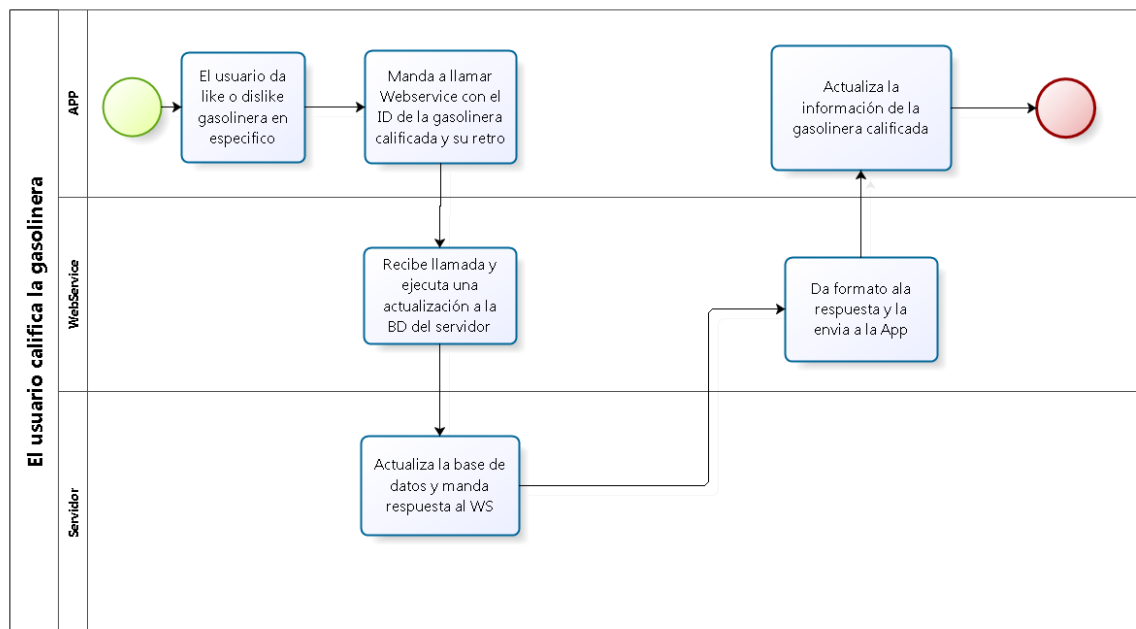


Fig. 12 Proceso de calificación del usuario a la gasolinera (Elaboración propia).

7.2.2 Arquitectura del software

La aplicación móvil funcionará en sistemas operativos Android 4.1 (Jelly Bean) y posteriores. Para comunicar la aplicación móvil con el servidor donde se encuentra la base de datos con la información de las gasolineras, se utilizarán *Web Services* (WS). Estos WS se utilizarán para cargar la lista de gasolineras y guardar su ranking, el servidor será Windows Server 2008, utilizando una base de datos de SQL Express y el desarrollo de los WS se hará en .NET.

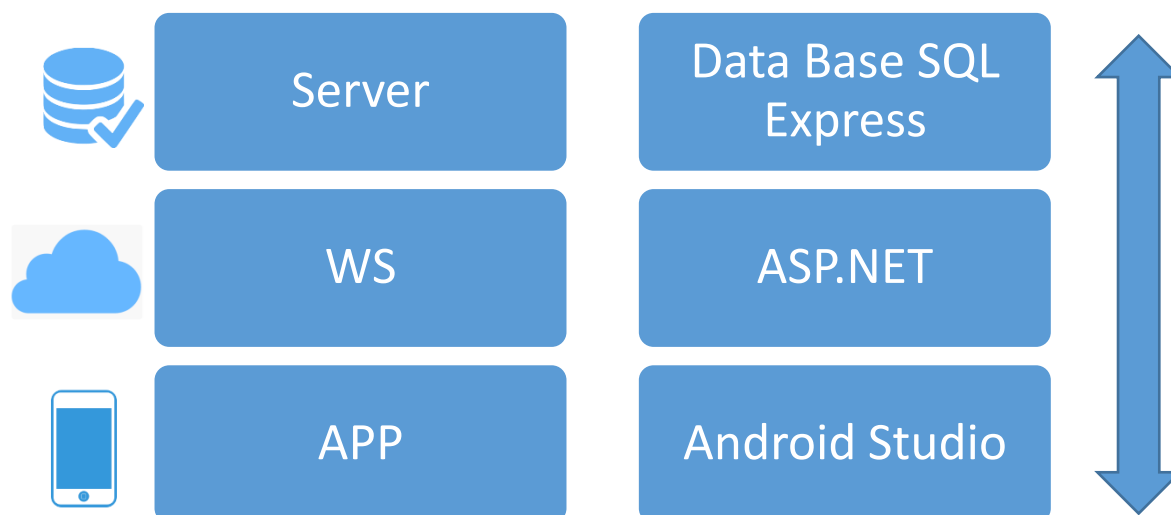


Fig. 13 Arquitectura de la app (Elaboración propia).

7.2.3 Diseño/plot de la secuencia de la App.

Se describe a grandes rasgos como es la secuencia de pantallas dentro de la App y una aproximación del diseño que debería tener cada una.

La aplicación iniciará en una pantalla principal que de entrada te muestra tu ubicación actual y te muestra todas las gasolineras que están al rededor por medio de un marcador con su color correspondiente en base a su calificación **Fig.14**:

Una vez estando dentro de la aplicación se elige alguna gasolinera y aparece una ventana de información **Fig.15**, de la cual al presionar te manda al detalle de la gasolinera, donde muestra datos de dirección y de sus calificaciones en *likes* y *dislikes* **Fig.16**, donde viene un botón para mostrar las opciones de dar *like* si se está de acuerdo con

lo que se cobró en la gasolinera y lo que se pagó, o dar *dislike* si no se está conforme, una vez hecho esto se actualiza la pantalla con la nueva calificación **Fig.17**.

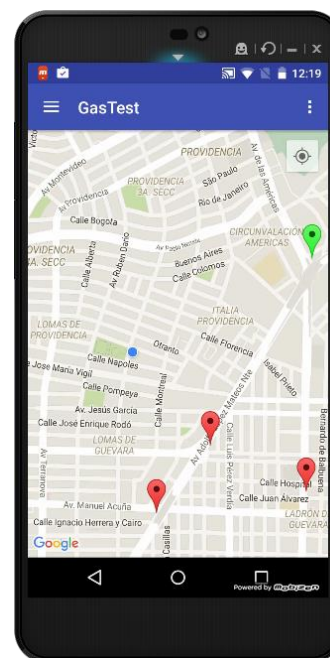


Fig. 14 Pantalla de inicio de la app

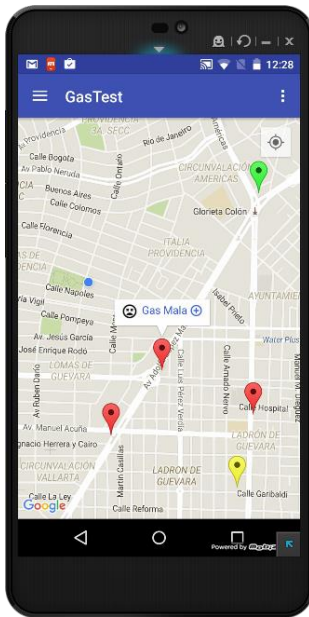


Fig. 15 Pantalla con calificación de la gasolinera



Fig. 16 Pantalla con el detalle de la gasolinera

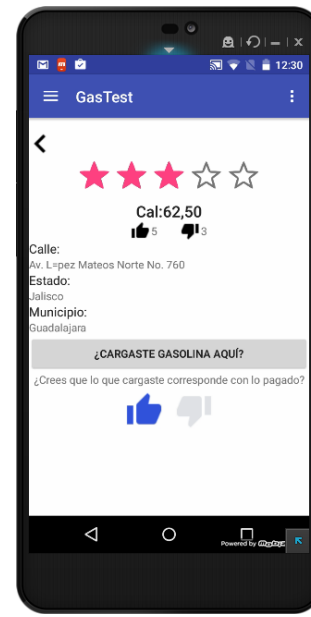


Fig. 17 Pantalla para actualizar calificación de gasolinera

Si le damos *back* y regresamos a la pantalla principal nos podemos dar cuenta que el semáforo de la gasolinera ya cambió de estatus, a uno amarillo **Fig. 18**. La aplicación tiene un menú principal (*drawer*) desde donde se acceder a otras opciones de la app **Fig. 19**.

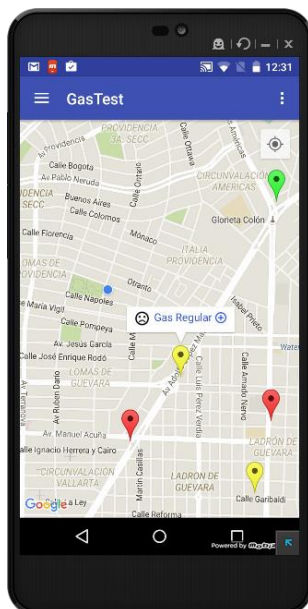


Fig. 18 Pantalla con gasolinera actualizada

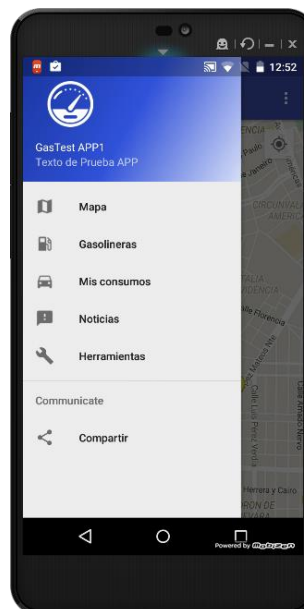


Fig. 19 Drawer con el menú de la app.

En la tabla `dbo.GAS_LIST` se encuentran los atributos de cada gasolinera, obtenidos del `gob.mx` (2017), estos atributos muestran información general de la gasolinera, los atributos principales son los de longitud y latitud, ya que estos van a ser utilizados por la aplicación para poner marcadores dentro de un mapa (Fig. 21).

ID	ESTADO	MUNICIPIO	DIRECCION	LATITUD	LONGITUD
1	E00003	Distrito Federal	Cuauhtemoc	Aldama No. 38	19.4402 -99.15187
2	E00004	Distrito Federal	Venustiano Carranza	H. Congreso De La Unión No. 22	19.45124 -99.11578
3	E00005	Distrito Federal	Gustavo A. Madero	Av. Instituto Politécnico Nacional No. 1881	19.49513 -99.13245
4	E00008	Distrito Federal	Miguel Hidalgo	Iturigaray No. 116 Y Vinreyes	19.42218 -99.2072
5	E00009	Distrito Federal	Benito Juárez	Calzada De Tlalpan No. 1395 Esq. Ajusco	19.36747 -99.14233
6	E00011	Distrito Federal	Benito Juárez	Alfonso XIII Esq. Obrero Mundial	19.40212 -99.14447
7	E00012	Distrito Federal	Benito Juárez	Av. Xola No. 18	19.39436 -99.13869
8	E00013	Distrito Federal	Ivvaro +bregon	Av. Revolución No. 1649	19.34644 -99.19034
9	E00014	Distrito Federal	Cuauhtemoc	MTnda No. 110	19.41905 -99.15728
10	E00016	Distrito Federal	Benito Juárez	Torres Adalid No. 1401	19.38929 -99.15725
11	E00017	Distrito Federal	Miguel Hidalgo	Av. Río San Joaquín No. 5976 Esq. Lago Ginebra	19.44554 -99.20139
12	E00019	Distrito Federal	Azcapotzalco	Av. De Las Granjas No. 751	19.49115 -99.1731
13	E00022	Distrito Federal	Miguel Hidalgo	Boulevard Manuel -vila Camacho No. 103	19.43196 -99.20892
14	E00026	Distrito Federal	Gustavo A. Madero	Av. Montevideo No. 424	19.49122 -99.13651
15	E00027	Distrito Federal	Miguel Hidalgo	Bahía De Santa Bárbara No. 182	19.43482 -99.17179
16	E00029	Distrito Federal	Ivvaro +bregon	Boulevard Adolfo López Mateos No. 282	19.38925 -99.19009
17	E00030	Distrito Federal	Benito Juárez	Caleta No. 28	19.40134 -99.14639
18	E00031	Distrito Federal	Cuauhtemoc	Barcelona No. 37 Esq. Versailles	19.42853 -99.1552

Fig. 21 Tabla `GAS_LIST` (Elaboración propia).

En la tabla `dbo.GAS_RANK` se encuentran la suma de *likes* y *dislikes* que los usuarios de la app han dado a las estaciones de servicio, dependiendo del botón que presionen en la app por medio de un *web service* hace un *update* a esta tabla (Fig. 22).

SQLQuery3.sql...istrator (52)*)

```
select * from dbo.GAS_RANK
```

ID	GOOD	BAD
1	E00003	0
2	E00004	0
3	E00005	0
4	E00008	0
5	E00009	0
6	E00011	0
7	E00012	0
8	E00013	0
9	E00014	0
10	E00016	0
11	E00017	0
12	E00019	0
13	E00022	0
14	E00026	0
15	E00027	0
16	E00029	0
17	E00030	0
18	E00031	0

Fig. 22 Tabla GAS_RANK (Elaboración propia).

La vista `dbo.V_GAS_LIST` une información de la tabla `dbo.GAS_LIST` (todo lo referente a la ubicación de las estaciones de servicio) con cuatro elementos que contienen la calificación de cada estación (Fig. 23), éstos vienen de la lista `dbo.V_GAS_RANK`. Esta vista es la que utiliza la app para cargar información, como una *best practice* no se conecta nunca a una tabla, si no a vistas de SQL.

SQLQuery4.sql...istrator (53)*)

```
USE [Test]
GO

/***** Object: View [dbo].[V_GAS_LIST]    Script Date: 03/20/2017 21:34:29 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER view [dbo].[V_GAS_LIST] as
select A.ID, A.ESTADO, A.MUNICIPIO, A.DIRECCION, A.LATITUD, A.LONGITUD, B.GOOD, B.BAD, B.PORC, B.FLAG
from GAS_LIST A INNER JOIN V_GAS_RANK B on B.ID = A.ID
GO
```

ID	ESTADO	MUNICIPIO	DIRECCION	LATITUD	LONGITUD	GOOD	BAD	PORC	FLAG
1	E00003	Distrito Federal	Cuauhtimoc	Aldama No. 38	19.4402	-99.15187	0	0	0
2	E00004	Distrito Federal	Venustiano Carranza	H. Congreso De La Unión No. 22	19.45124	-99.11578	0	0	0
3	E00005	Distrito Federal	Gustavo A. Madero	Av. Instituto Politécnico Nacional No. 1881	19.49513	-99.13245	0	0	0
4	E00008	Distrito Federal	Miguel Hidalgo	Itumbary No. 116 Y Virreyes	19.42218	-99.2072	0	0	0
5	E00009	Distrito Federal	Benito Juárez	Calzada De Tlalpan No. 1395 Esq. Ajusco	19.36747	-99.14233	0	0	0
6	E00011	Distrito Federal	Benito Juárez	Alfonso Xili Esq. Obrero Mundial	19.40212	-99.14447	0	0	0
7	E00012	Distrito Federal	Benito Juárez	Av. Xola No. 18	19.39436	-99.13869	0	0	0
8	E00013	Distrito Federal	Ivaro + bregon	Av. Revolución No. 1649	19.34644	-99.19034	0	0	0
9	E00014	Distrito Federal	Cuauhtimoc	MTnda No. 110	19.41905	-99.15728	0	0	0
10	E00016	Distrito Federal	Benito Juárez	Torres Adalid No. 1401	19.38929	-99.15725	0	0	0
11	E00017	Distrito Federal	Miguel Hidalgo	Av. Rto San Joaquín No. 5976 Esq. Lago Ginebra	19.44554	-99.20139	0	0	0
12	E00019	Distrito Federal	Azcapotzalco	Av. De Las Granjas No. 751	19.49115	-99.1731	0	0	0

Fig. 23 Vista V_GAS_LIST (Elaboración propia).

En la vista `dbo.V_GAS_RANK` (Fig. 24) se hacen los cálculos para determinar la calificación de cada gasolinera, de acuerdo con el marco teórico existen diferentes métodos para determinar una calificación de reputación de alguna entidad, en este caso el método utilizado es un porcentaje de las calificaciones positivas sobre el total de las calificaciones, siendo 100% la calificación más alta y 0% la más baja. Además la terminación de un *flag*, que va ayudar a que dentro de la aplicación móvil se pueda determinar qué color de marcador le debe poner a cada gasolinera, si el porcentaje de *likes* sobre el total de calificaciones va de 80% a 100% el indicador es igual a tres y en la aplicación se pintara como verde, si el porcentaje es menor que 80% pero mayor que 60% el indicador será dos y en la aplicación se pintara como amarillo, si el porcentaje es menor a 60% el indicador es igual a 3 y la app lo pintará como rojo, en caso de que no exista ninguna calificación el indicador será igual a 0 y en la app se mostrará el marcador transparente.

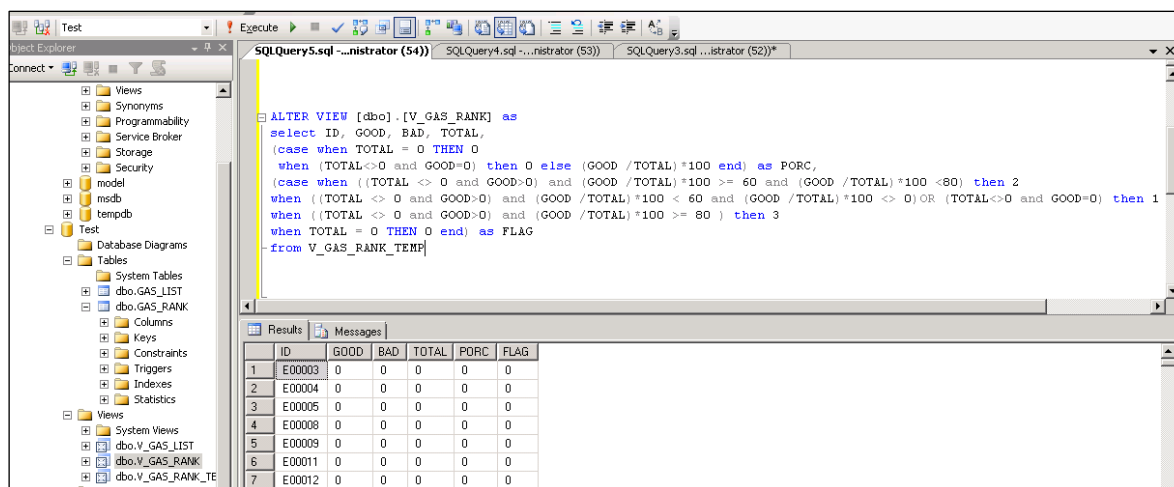


Fig. 24 Vista `V_GAS_RANK` (Elaboración propia).

Los datos de las calificaciones sin cálculo vienen de una vista intermedia llamada `dbo.V_GAS_RANK_TEMP` (Fig. 25).

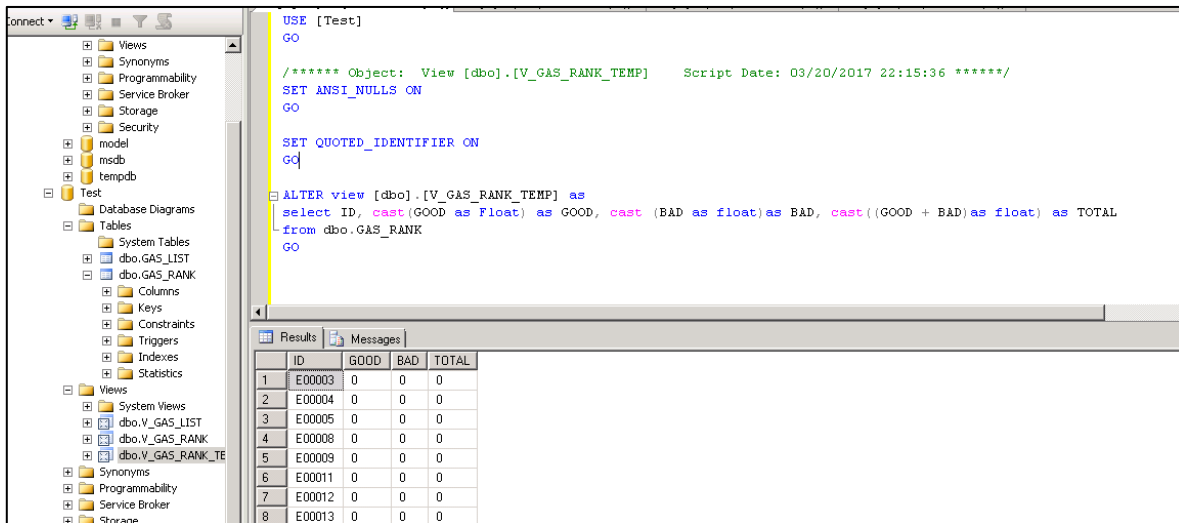


Fig. 25 Vista V_GAS_RANK_TEMP (Elaboración propia).

El *web service* que se creó cuenta con cuatro métodos (Fig. 26), estos métodos se encargan de la comunicación de la app y la base de datos, para poder extraer y/o insertar desde la base de datos:

- ShowAllGas: Trae la lista de todas las gasolineras con su calificación (se utiliza al inicio de la aplicación al cargar el mapa, y siempre que se regresa a la pantalla principal).
- MarkerDetail: Se utiliza cuando se ingresa a una Gasolinera en específico, trae información sobre la dirección de la gasolinera y la calificación con el detalle de sus *likes* y *dislikes*.
- LikePressed: Actualiza el *like* en la base de datos, sumándole uno al registro ya existente.
- DislikePressed: Actualiza el *Dislike* en la base de datos, sumándole uno al registro de ya existente.

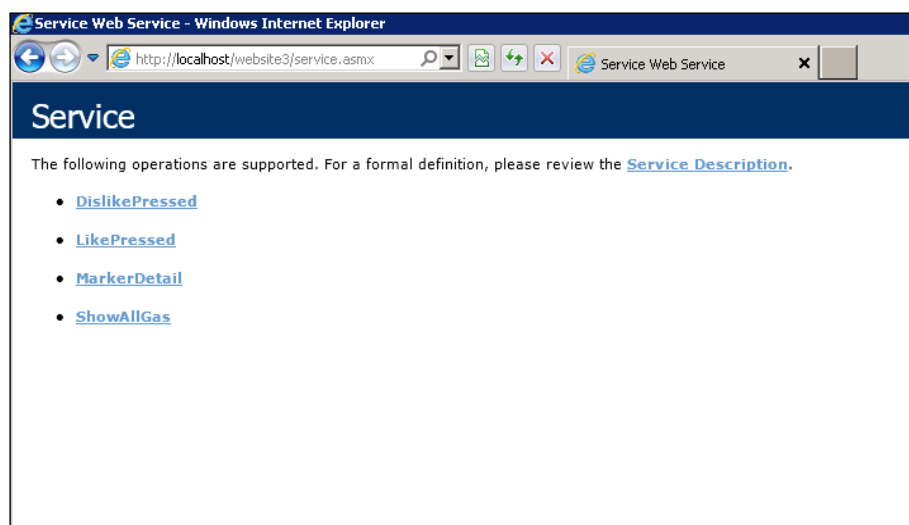


Fig. 26 Web Service (Elaboración propia).

7.2.2 Creación de pantallas y codificación de la App.

La aplicación fue desarrollada en la plataforma Android Studio, que utiliza como lenguaje de programación java. Antes de crear las clases que son las que contienen todas las instrucciones del funcionamiento de la App es necesario crear los elementos de las pantallas en XML, esta app consta de 7 recursos XML para pantallas y dos recursos más de menú (Fig. 27).

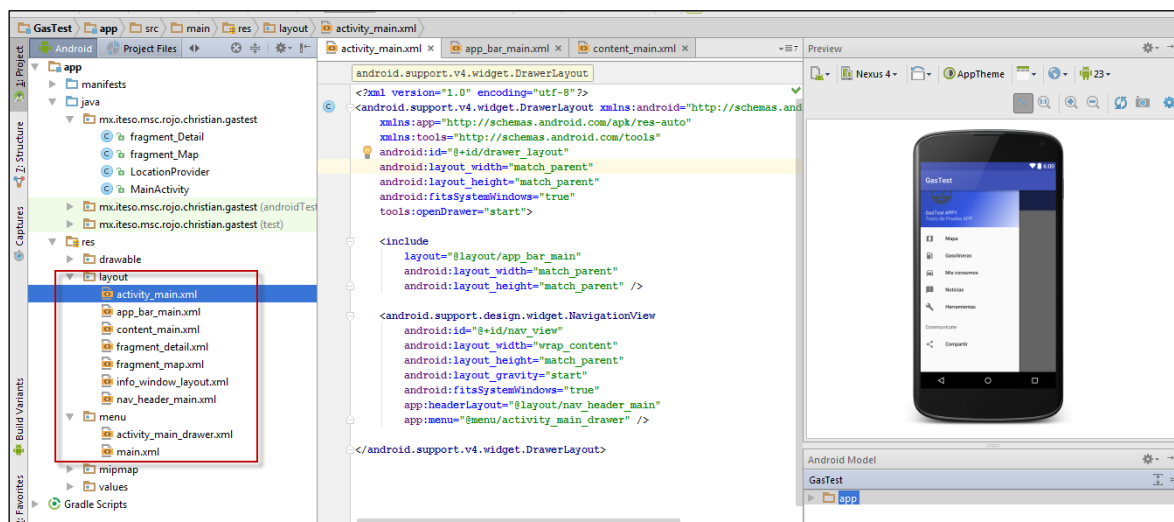


Fig.27 Plataforma Android Studio (Elaboración propia).

Los recursos XML de tipo *layout* son los siguientes:

- **activity_main.xml:** Este *layout* contiene un elemento de tipo *drawer*, contiene el menú *activity_main_drawer.xml*, que viene siendo el menú principal de la aplicación, y es desde donde se puede acceder a las diferentes pantallas de la app.
- **app_bar_main.xml:** Es un contenedor únicamente, que va a tener dentro el recurso *content_main.xml*.
- **content_main.xml:** Este elemento es un contenedor, que contiene un *fragment* que se va cargar dependiendo de lo que elija el usuario en el menú principal.
- **fragment_detail.xml:** Contiene un *fragment* con los elementos que se van a mostrar en la pantalla cuando un usuario elija una gasolinera, y desde aquí es donde se va a calificar ésta (Fig. 28)

- `fragment_map.xml`: Este *layout* contiene un *fragment* que carga una mapa desde la api de Google maps, y en este mapa es donde se insertan los marcadores de la ubicación de las gasolineras a la hora de estarse cargando.
- `info_window_layout.xml`: Éste servirá para que se muestre una pequeña ventana cuando se le de *click* a algún marcador dentro del mapa.
- `nav_header_main.xml`: Este recurso contiene el icono de la app y la descripción de la pantalla en la cual se está.

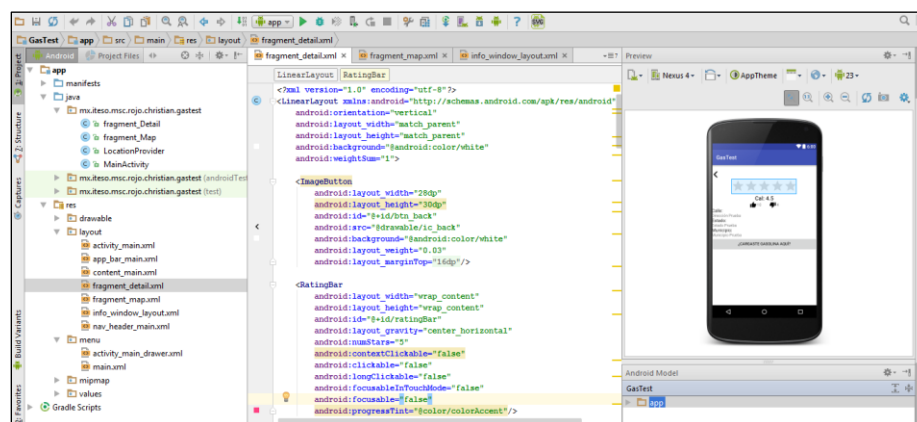


Fig. 28 Layout `fragment_detail.xml` (Elaboración propia).

Como se aprecia en la Fig. 27 esta App consta de 4 clases, comenzando con una *main activity*, que para java es la clase principal desde donde inicia nuestra aplicación y se manda a llamar las diferentes pantallas.

En la clase *MainActivity* se manda a llamar la clase *LocationProvider* (Fig. 29), que es la que va a determinar la ubicación del usuario de la app (a través de su GPS) y con esta información lo va a ubicar en un mapa de Google maps donde le mostrará las gasolineras a su alrededor ya con su información actualizada, esto lo hace por medio de llamada a la función de *web service* *ShowAllGas*, que ya fue descrita anteriormente (Fig. 30).

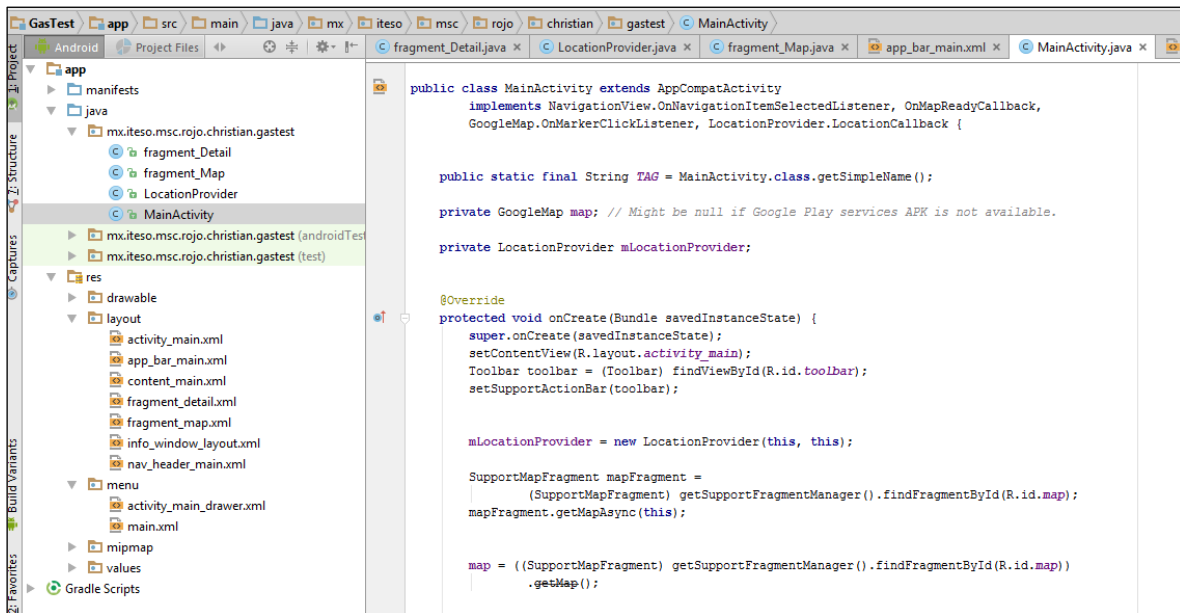


Fig. 29 Clase MainActivity (Elaboración propia).

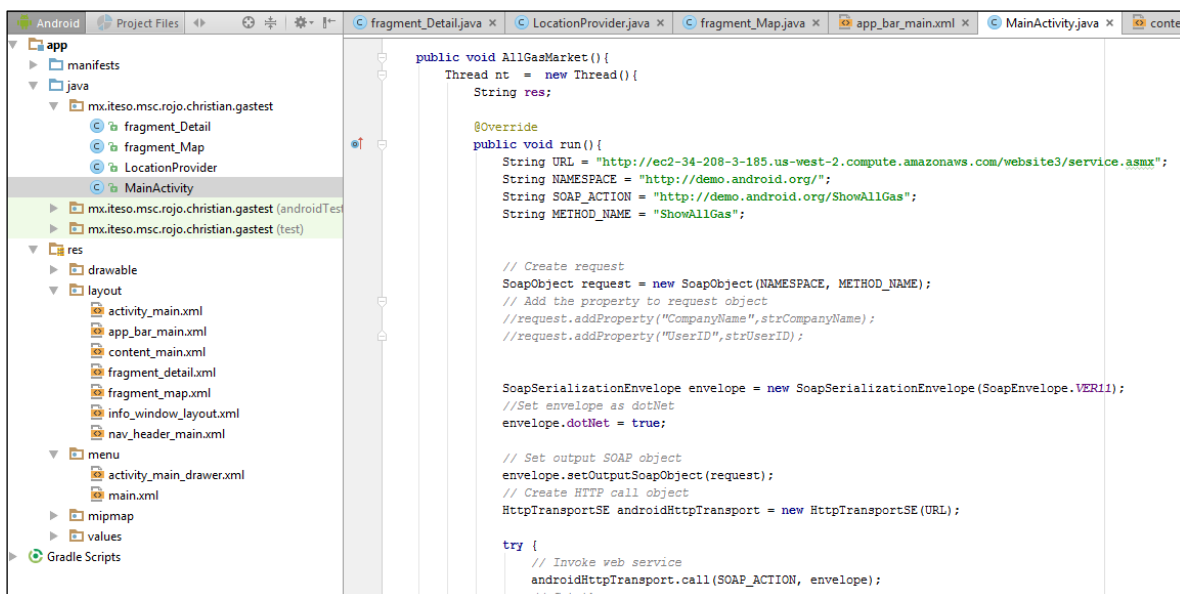


Fig. 30 Ejecución de WS ShowAllGas en la clase MainActivity (Elaboración propia).

La clase fragment_Map se encarga únicamente de llenar el elemento de xml que va a contener el mapa de Google maps, los marcadores se definen y se colorean desde la actividad principal

La clase fragment_Detail (Fig. 31) es la que se encarga de darle el funcionamiento al fragment_detail.xml, que es la pantalla que se le va a mostrar al

usuario cuando de elija una gasolinera en particular, y desde aquí va a poner su calificación a esta gasolinera, por medio de los WS LikePressed y DislikePressed.

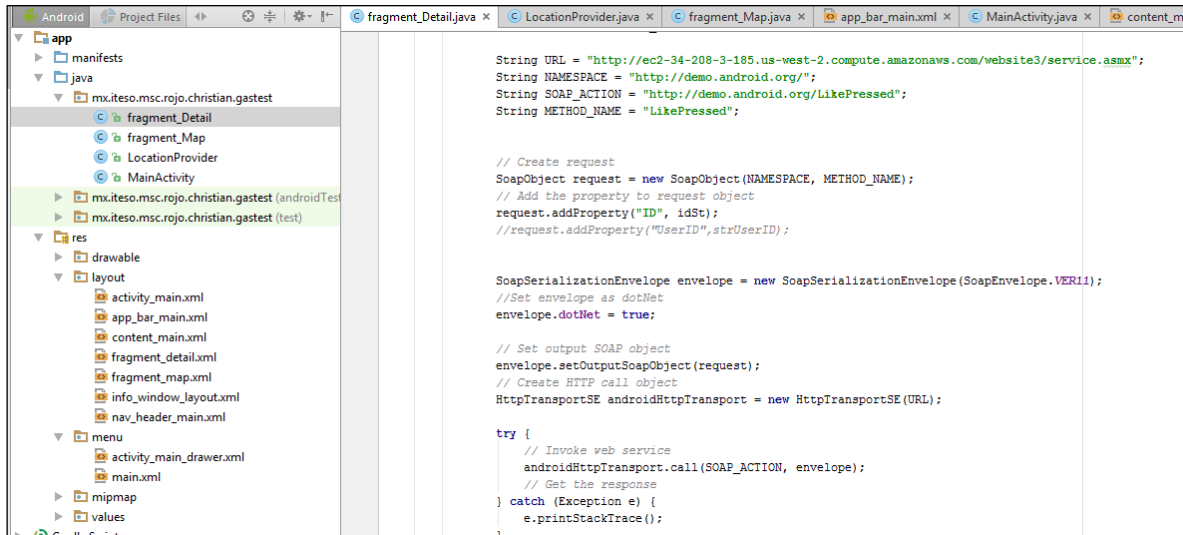


Fig. 31 Clase fragment_Detail (Elaboración propia).

7.4 PRUEBAS.

7.4.1 Pruebas Unitarias.

Estas son las primeras pruebas que se hicieron por parte del mismo usuario desarrollador de la aplicación para verificar el correcto funcionamiento de la app y sus demás componentes.

Prueba	Descripción	Resultado esperado	Estatus	Usuario	Comentarios
1	LoadMap	La aplicación carga un mapa de Google maps en la pantalla	100	Christian Rojo	Requiere de la API de Google Maps.
2	LocationProvider	La aplicación muestra la ubicación real del usuario en el momento de utilizar la app.	100	Christian Rojo	Requiere de permisos por parte del usuario, para que la app pueda acceder a su ubicación
3	Llamado de WS ShowAllGas	En la aplicación se muestran los marcadores con la ubicación correcta de las gasolineras cercanas.	100	Christian Rojo	Se requiere de acceso a internet.
4	Semaforización de los marcadores	Los marcadores donde se ubican las gasolineras se muestran de color dependiendo de la calificación que tienen éstas en la base de datos.	100	Christian Rojo	Este funcionamiento requiere que la ejecución del WS ShowAllGas haya sido correcto.
5	Abre ventana rápida de marcador	Al posicionarse sobre un marcador muestra una ventana con una	100	Christian Rojo	Este funcionamiento requiere que la ejecución del WS ShowAllGas haya sido correcto.

		descripción rápida de su calificación.			
6	Abre pantalla con el detalle de la gasolinera	Al darle click a una gasolinera en específico nos muestra su información de ubicación y su calificación actualizada.	100	Christian Rojo	Este funcionamiento requiere de la ejecución del WS MarkerDetail.
7	Click al botón de Like	Al presionar la mano de Like dentro de la pantalla del detalle de la gasolinera se manda a llamar el WS LikePressed	100	Christian Rojo	Se necesita de internet.
8	Llamado de WS LikePressed	El WS actualiza la base de datos sumando una a la columna de good en la tabla GAS_RANK	100	Christian Rojo	
9	Click al botón de Dislike	Al presionar la mano de Dislike dentro de la pantalla del detalle de la gasolinera se manda a llamar el WS DisikePressed	100	Christian Rojo	Se necesita de internet.
10	Llamado de WS DisikePressed	El WS actualiza la base de datos sumando una a la columna de bad en la tabla GAS_RANK	100	Christian Rojo	
11	Actualización de la calificación después de la retro del usuario.	Se vuelve a ejecutar el WS MarkerDetail, y la nueva calificación se muestra en pantalla.	100	Christian Rojo	Es necesario que alguno de los WS que se haya elegido Like o Dislike se haya ejecutado correctamente. Y que el llamado de MarkerDetail se vuelva a dar.

13	Abre Menú	Al hacer click sobre el icono de menú lo abre.	100	Christian Rojo	
14	Cierra Menú	Con el menú abierto, al hacer click en el icono de menú o en alguna opción de este se cierra	100	Christian Rojo	

7.4.2 Pruebas Integrales.

Estas pruebas se hicieron por parte usuarios ajenos al desarrollador de la aplicación para verificar el funcionamiento de la app.

Prueba	Descripción	Resultado esperado	Estatus	Usuario	Comentarios
1	Abrir la aplicación con su localización	La aplicación carga un mapa de Google maps en la pantalla y muestra las gasolineras al cercanas a la ubicación del usuario.	100	Antonio Arredondo	Requiere de permisos por parte del usuario, para que la app pueda acceder a su ubicación
2	Navegación dentro del mapa	La pantalla responde correctamente a los movimientos con los dedos por parte del usuario (zoom, desplazamiento, etc.)	100	Antonio Arredondo	Requiere que el usuario esté familiarizado con los gestos básicos de pantallas <i>touch</i> .
3	Ventana rápida con información en específico de una Gasolinera.	Al darle click sobre una gasolinera abre una ventana con información rápida del estatus de esa gasolinera.	100	Antonio Arredondo	Requiere que el usuario esté familiarizado con los gestos básicos de pantallas <i>touch</i> .

4	Pantalla con información de gasolinera seleccionada, lista para cargar la calificación otorgada por el usuario.	Muestra pantalla con dirección y datos de ubicación de la gasolinera seleccionada, y desde esta pantalla es posible que el usuario ponga su calificación.	100	Antonio Arredondo	
5	Actualización de calificación positiva por parte del usuario.	El usuario puede calificar y subir su calificación desde la app.	100	Antonio Arredondo	Se necesita de internet.
6	Actualización de calificación negativa por parte del usuario.	El usuario puede calificar y subir su calificación desde la app.	100	Antonio Arredondo	Se necesita de internet.
7	Funcionamiento general.	La aplicación no tiene cierres o reinicios inesperados.	100	Antonio Arredondo	

7.5 LIBERACIÓN DEL APK/RESULTADOS.

El APK de esta app se compartió con 30 usuarios de la zona metropolitana de Guadalajara a los cuales se les solicitó que llenaran una encuesta antes de usar la aplicación y otra posterior al uso de la aplicación (el formato de la encuesta viene en la parte de anexos).

En la primera encuesta se le hicieron preguntas relacionadas a sus hábitos de consumo en materia de combustibles y si creían que en las gasolineras despachaban la cantidad de combustible que cobraban, a lo cual un 96.7% de los encuestados respondió que no (Fig. 32).

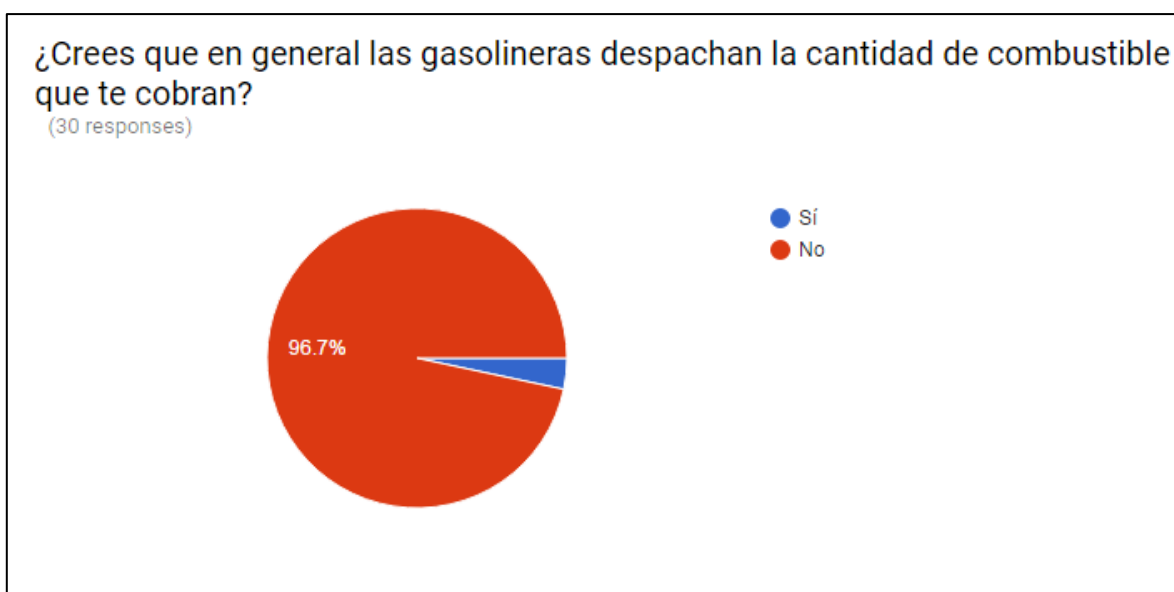


Fig. 32 Resultado a pregunta en encuesta inicial (Elaboración propia desde formularios de Google).

Un 80% de los encuestados respondió que tenía una gasolinera preferida, y de estos en un 54.2% reveló que su preferencia era por motivos de confianza en lo que se les despachaba (Fig. 33).

El 100% de los encuestados contestó que si existiera una base con información sobre la reputación de las gasolineras de su localidad la consultaría, y un 93.3% estaría dispuesto a cooperar para que esta base de datos estuviera actualizada.



Fig. 33 Resultado a pregunta en encuesta inicial (Elaboración propia desde formularios de Google).

Posteriormente al uso de la aplicación se realizó otra encuesta, donde se les hacía a los usuarios las siguientes preguntas:

- ¿Te fue sencillo dar tu retroalimentación sobre la gasolinera seleccionada?
Un 90% de los usuarios contestó que le fue sencillo dar esta retroalimentación.
- ¿Consultarías esta app para conocer la reputación de las gasolineras de tu localidad?
Un 96.7% de los usuarios respondió que sí utilizaría esta app.
- ¿Las calificaciones de esta base de datos podrían modificar tu preferencia habitual de gasolinera?
Para un 90% de los usuarios las calificaciones de esta base podrían modificar su preferencia de gasolinera.

- ¿Cuál consideras como la principal ventaja de que esta base de datos se encuentre dentro de una app?

Las respuestas más mencionadas fueron: portabilidad, disponibilidad, facilidad de acceso y localización en tiempo real.

- ¿Cuál consideras como la mayor desventaja de que esta base de datos se encuentre dentro de una app?

Las respuestas más mencionadas fueron: consumo de datos de internet, dependencia de un smartphone, no todas las personas saben manejar apps y limitada a los recursos del dispositivo móvil.

- ¿Estarías dispuesto a seguir cooperando para tener actualizada esta base de datos?

El 93.3% de los encuestados mencionó que sí.

8. CONCLUSIONES.

Este trabajo de obtención de grado tuvo como principal objetivo el “Creación de un sistema de reputación sobre gasolineras en México, utilizando una aplicación móvil, para que los usuarios puedan compartir sus experiencias y valoraciones al cargar gasolina en una estación en específico”.

Para lograr el correcto funcionamiento de este sistema fue necesaria la creación de tres elementos:

1. Base de datos en SQL con la información de las gasolineras en México.
2. Una aplicación desarrollada en sistema operativo Android.
3. Un servicio web que logra la comunicación entre estos dos elementos.

Los objetivos particulares de este proyecto fueron:

- Permitir a los usuarios tomar una mejor decisión a la hora de elegir una estación donde cargar gasolina, gracias a la base de información que los mismos usuarios estarían generando.

Esto se logra permitiendo a los usuarios subir su calificación después de cargar en alguna estación de servicio, y haciendo pública esta información a través de la app, para que los usuarios que la utilicen puedan ver esta base de datos con las calificaciones por gasolineras de los demás usuarios.

- Exponer a las estaciones de servicio que estén otorgando un mal servicio o los usuarios consideren que no están entregando la cantidad de combustible que están cobrando, y de esta manera forzarlos a que se regularicen.

Este objetivo se lograría una vez que la aplicación se publique dentro del *Marketplace* de google, ya que actualmente queda como un supuesto.

- Conocer el sentir de los usuarios después de utilizar la aplicación, mediante la generación de un reporte de resultados de las pruebas para ver si se cumple con los objetivos propuestos.

Como se puede apreciar en el punto 7.5 de este trabajo los usuarios después de utilizar este prototipo de app seguían con intención de uso del mismo, además de que creían que la base de datos generada por la retroalimentación de todos los usuarios les podía ser útil para tomar una decisión sobre en qué gasolinera cargar combustible.

Por lo tanto se puede decir que sí se cumple con la hipótesis propuesta, que por medio de una base de datos visible para los usuarios que contenga la información de reputación de gasolineras en México puede ayudar a que el usuario tome una mejor decisión a la hora de cargar gasolina con determinada estación de servicio, o que al menos se sienta más seguro cuando lo hace.

De proyecto de obtención de grado yo quería realizar un trabajo que no se quedara solo conmigo, si no que pudiera ayudar a la sociedad de alguna manera, es por eso que estuve buscando proyectos que tuvieran algún impacto positivo en la sociedad actual.

Como ya se mencionó en la introducción y justificación el tema de los combustibles en México tiene un gran impacto en la economía de todo el país

(actualmente también está teniendo influencia más allá de lo económico, llegando a tener tintes políticos), es por ello necesario encontrar medios para que la calidad de los servicios de combustible en México mejore.

Este trabajo tiene elaborándose desde finales de 2015 por lo que durante este tiempo han cambiado algunas cosas con respecto al contexto con el que se hizo la propuesta de este trabajo:

- El precio de las gasolinas que se daba mensualmente por parte del gobierno, ya no es fijo en todo México, puede cambiar diariamente y ser distinto dependiendo el municipio.
- Apertura en materia de comercializadores de combustibles, PEMEX ya no es la única empresa que puede vender gasolina y diésel en México, otras marcas compañías pueden hacerlo.
- Salida al mercado de aplicaciones que tienen prácticamente las mismas funciones que las que este trabajo propone.

Este proyecto me fue de gran aprendizaje ya que no había estado en contacto con algún proyecto de desarrollo en Java y tampoco con el desarrollo de una aplicación móvil. Por lo que tuve aprender este lenguaje y sus aplicaciones en el entorno de una app de Android, lo cual me llevó más tiempo del esperado el desarrollo del código de la aplicación, que el que le hubiera llevado a un programador con más experiencia. Pude comprobar que aunque parezca sencillo o accesible el hacer una app (debido a que no implica una inversión fuerte) es todo un proyecto de ingeniería de Software, y si realmente se quiere hacer una app de manera profesional y que pueda ser rentable es necesario el invertirle tiempo y recursos para su correcto funcionamiento.

Para la elaboración de este trabajo tuve que aplicar varios de los conocimientos aprendidos durante las diversas asignaturas que lleve durante la maestría como: administración de proyectos, metodología de procesos, sistemas de información gerenciales, desarrollo de aplicaciones móviles, manejo y análisis

de información masiva, entre otras. Estas asignaturas me enriquecieron profesionalmente y en mi actividad laboral diaria.

Para este prototipo de aplicación lo que estaría pendiente es afinar el tema de los precios por municipio dentro del detalle de cada gasolinera (actualmente lo tiene a nivel general), mejorar el diseño de las pantallas, encontrar como generar una ventaja competitiva o diferenciador sobre las aplicaciones que existen actualmente y subirla al Marketplace de Google para su uso.

BIBLIOGRAFÍA.

- Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., & Fowler, M. (2001). Manifesto for agile software development. *Utah: The Agile Alliance*.
Recuperado a partir de <http://agilemanifesto.org/> consultado el 20 de Abril de 2016.
- Cuello, J., & Vittone, J. (2013). Diseñando apps para móviles. Recuperado a partir de <http://appdesignbook.com/> consultado el 17 de Abril de 2017.
- Duan, W., Gu, B., & Whinston, A. B. (2008). Do online reviews matter? — An empirical investigation of panel data. *Decision Support Systems*, 45(4), 1007–1016.
- IEEE. (1990). IEEE standard glossary of software engineering terminology. *IEEE Standard Board*.
- INEGI. (2014). *Vehículos de motor registrados en circulación* [BD INEGI]. Recuperado a partir de http://www.inegi.org.mx/sistemas/olap/Proyectos/bd/continuas/transporte/vehiculos.asp?s=est&c=13158&proy=vmrc_vehiculos consultado el 7 de Julio de 2016.
- Jøsang, A., & Golbeck, J. (2009, septiembre). Challenges for robust trust and reputation systems. *5th International Workshop on Security and Trust Management*.
- Jøsang, A., Ismail, R., & Colin, B. (2007). A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(Emerging Issues in Collaborative Commerce), 618-644.
- Koutrouli, E., & Tsalgatidou, A. (2015). Reputation systems evaluation survey. *ACM Computing Surveys*, 48(3), p28.
- Ma, X. (Sean), Khansa, L., Deng, Y., & Kim, S. S. (2014). Impact of prior reviews on the subsequent review process in reputation systems. *Journal of Management Information Systems*, 30(3), 279–310.

-
- Mousa, H., Mokhtar, S. B., Hasan, O., Younes, O., Hadhoud, M., & Brunie, L. (2015). Trust management and reputation systems in mobile participatory sensing applications: A survey. *Computer Networks*, 90, 49-73.
- Nayebi, F., Desharnais, J.-M., & Abran, A. (2012). The state of art of mobile application usability evaluation. *Canadian Conference on Electrical and Computer Engineering*.
- Parera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Sensing as a service model for smart cities supported by Internet of Things. *Transactions on Emerging Telecommunications Technologies*, 00, 1-12.
- PEMEX. (2014). Anuario Estadístico 2014. Recuperado a partir de http://www.pemex.com/ri/Publicaciones/Anuario%20Estadistico%20Archivos/2014_ae_00_vc_e.pdf consultado el 04 de Mayo de 2016.
- Pressman, R. (2010). *Ingeniería del software. Un enfoque práctico* (7 ma). McGraw - Hill.
- Prieto Blázquez, J., Ramírez Vique, R., Morillo Pozo, J. D., & Domingo Prieto, M. (2011). *Tecnología y desarrollo en dispositivos móviles*. Barcelona: Fundació per a la Universitat Oberta de Catalunya.
- PROFECO. (2016). *Programa Nacional de Verificación de Combustibles “¿Quién es quién en los combustibles?”*. Recuperado a partir de <http://200.53.148.113/qgg/> consultado el 08 de Agosto de 2016.
- Ramírez Vique, R. (2013). *Métodos para el desarrollo de aplicaciones móviles* (Universitat Oberta de Catalunya). Barcelona.
- Resnick, P., & Zeckhauser, R. (2002). Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. *Advances in Applied Microeconomics: A Research Annual*, 11.
-

Resnick, P., Zeckhauser, R., Friedman, E., & Kuwabara, K. (2012). Reputation systems.

Communications of the ACM, 43(12), p45-48.

Secretaría de Economía. Instrumentos de medición-Sistema para medición y despacho de gasolina y otros combustibles líquidos-Especificaciones, métodos de prueba y de verificación., NOM-005-SCFI-2011 NOM-005-SCFI-2011 § (2012). Recuperado a partir de http://dof.gob.mx/nota_detalle.php?codigo=5241707&fecha=30/03/2012 consultado el 20 de Agosto de 2016.

Shankar, P., Huang, W., Castro, P., Nath, B., & Iftode, L. (2012). Crowds replace experts: Building better location-based services using mobile social network interactions.

IEEE International Conference on Pervasive Computing & Communications, 2012(2).

Sommerville, I. (2005). *Ingeniería del software* (7 ma). Pearson Educación.

Wangenheim, F. v., & Bayón, T. (2007). The chain from customer satisfaction via word-of-mouth referrals to new customer acquisition. *Academy of Marketing Science*, 35(2), p233–249.

ANEXOS.

Encuesta inicial de gasolineras:

Nombre: *

Tu respuesta

¿Qué tipo de combustible utilizas? *

- ☐ Gasolina Magna
- ☐ Gasolina Premium
- ☐ Diesel

¿Cuánto cargas en promedio por semana (en pesos)? *

Tu respuesta

¿Crees que en general las gasolineras despachan la cantidad de combustible que te cobran? *

- ☐ Sí
- ☐ No

¿Tienes alguna gasolinera preferida? *

- ☐ Sí
- ☐ No

En caso de haber contestado sí ¿Por qué razón es tu preferida?

- ☐ Confianza en lo que despachan
- ☐ Ubicación
- ☐ Rapidez en el servicio
- ☐ Facilidad de pago
- ☐ Otro:

¿Si existiera alguna base de datos con la información de la reputación de las gasolineras de tu localidad la consultarías? *

- ☐ Sí
- ☐ No

¿Estarías dispuesto a cooperar con tu retroalimentación para actualizar esta base de datos? *

- ☐ Sí
- ☐ No

ENVIAR

Encuesta después de uso de la aplicación:

Nombre: *

Tu respuesta

¿Te fue sencillo dar tu retroalimentación sobre la gasolinera seleccionada? *

☐ Sí

☐ No

¿Consultarías esta app para conocer la reputación de la gasolineras de tu localidad? *

☐ Sí

☐ No

¿Las calificaciones de esta base de datos podrían modificar tu preferencia habitual de gasolinera? *

☐ Sí

☐ No

¿Cuál consideras como la principal ventaja de que esta base de datos se encuentre dentro de una app? *

Tu respuesta

¿Cuál consideras como la mayor desventaja de que esta base de datos se encuentre dentro de una app? *

Tu respuesta

¿Estarías dispuesto a seguir cooperando para tener actualizada esta base de datos? *

☐ Sí

☐ No

ENVIAR